



LIRE ET ÉCRIRE DANS UN FICHIER, FICHIERS CSV



POUR OUVRIR / CRÉER / RENOMMER  
/ DÉPLACER / SUPPRIMER  
DES **DOSSIERS**

ET POUR OUVRIR / CRÉER / MODIFIER  
/ DÉPLACER / SUPPRIMER  
DES **FICHIERS**

IL EXISTE LA BIBLIOTHÈQUE  
« **System.IO** »



# LA MANIPULATION DE DONNÉES ET DE FICHIERS EST UNE COMPOSANTE ESSENTIELLE DES LOGICIELS

- LA PLUPART DES LOGICIELS ONT POUR BUT DE TRAITER DES DONNÉES :
  - CERTAINS D'ENTRE EUX ONT DES DONNÉES D'ENTRÉES QUI PROVIENNENT DE FICHIERS
  - D'AUTRES ONT DES DONNÉES DE SORTIES QUI SONT DESTINÉES À ALLER DANS DES FICHIERS
- LORSQU'UN LOGICIEL GÉNÈRE UN FICHIER LOG : IL ÉCRIT DANS UN FICHIER
- LORSQU'UN LOGICIEL CHARGE UN FICHIER (VIA LA FONCTIONNALITÉ « OUVRIR »), IL LIT UN FICHIER
- LES FICHIERS PEUVENT ÊTRE BRUTS, À LARGEUR FIXE, CSV, XML, JSON OU AUTRES



# SAVOIR MANIER LES CHEMINS D'ACCÈS

- LE CHEMINS D'ACCÈS SONT COMPOSÉS D'ANTI-SLASH « \ ».
- AFIN D'ÉVITER LES ERREURS, GÉNÉRER UN CHEMIN D'ACCÈS EN FAISANT DES CONCATÉNATIONS
- IL FAUT UTILISER LA CLASSE « `Path` » :
  - « `Path.GetDirectoryName(@"C:\Users\Yohan\Bureau\Fichier.txt")` »  
OBTIENT LE CHEMIN COMPLET DU DOSSIER (DONC « C:\USERS\YOHAN\BUREAU »)
  - À L'INVERSE, « `Path.GetFileName(@"C:\Users\Yohan\Bureau\Fichier.txt")` »  
OBTIENT LE NOM COMPLET DU FICHIER, SANS LE CHEMIN (DONC « FICHIER.TXT »)
  - ET SURTOUT : « `Path.Combine(@"C:\Users", "Yohan", "Bureau", "Fichier.txt")` »  
RETOURNE « C:\USERS\YOHAN\BUREAU\FICHIER.TXT »

**Note : L'« @ » devant une chaîne en dur permet au « \ » d'être considéré comme un caractère normal (et non comme une séquence d'échappement)**

# VOICI QUELQUES EXEMPLES DE MANIPULATION DE DOSSIERS

SAVOIR SI UN DOSSIER EXISTE : **Directory.Exists** (@ "D : \Sauvegarde" ) ;

CRÉER UN DOSSIER : **Directory.CreateDirectory** (@ "D : \Sauvegarde" ) ;

RENOMMER UN DOSSIER : **Directory.Move** (@ "D : \Sauvegarde" , "D : \Sauvegardes" ) ;

SUPPRIMER UN DOSSIER : **Directory.Delete** (@ "D : \Sauvegarde" ) ;

LISTER LES FICHIERS D'UN DOSSIER (NON-RÉCURSIF) : **Directory.GetFiles** (@ "D : \Sauvegardes" ) ;

LISTER LES FICHIERS D'UN DOSSIER (RÉCURSIF) :

**Directory.GetFiles** (@ "D : \Sauvegardes" , "\*" . "\*" , SearchOption.AllDirectories) ;



# VOICI QUELQUES EXEMPLES DE MANIPULATION DE FICHIERS

- ÉCRIRE DU TEXTE DANS UN FICHIER :

```
File.WriteAllText(@"C:\Fichier1.txt", "Texte...");
```

- LIRE TOUTES LES DONNÉES CONTENUES DANS UN FICHIER :

```
string donnees = File.ReadAllText(@"C:\Fichier2.txt");
```

- METTRE TOUTES LES LIGNES CONTENUES DANS UN FICHIER ET METTRE DANS UN TABLEAU :

```
string[] lignes = File.ReadAllLines(@"C:\Fichier3.txt");
```

- COPIER UN FICHIER :

```
File.Copy(@"C:\Fichier4.txt", @"C:\Dossier\Fichier4.txt");
```

**Attention : si le fichier est de grande taille (plus de 25 Mo), une lecture par flux est préférable (StreamReader, StreamWriter)**

```
1 Car;MPG;Cylinders;Displacement;Horsepower;Weight;Acceleration;Model;Origin
2 Peugeot 504;25.0;4;110.0;87.00;2672.;17.5;70;Europe
3 Audi 100 LS;24.0;4;107.0;90.00;2430.;14.5;70;Europe
4 BMW 2002;26.0;4;121.0;113.0;2234.;12.5;70;Europe
5 Ford F250;10.0;8;360.0;215.0;4615.;14.0;70;US
6 Toyota Corolla;25.0;4;113.0;95.00;2228.;14.0;71;Japan
7 Ford Pinto;25.0;4;98.00;0;2046.;19.0;71;US
8 Volkswagen Super Beetle 117;0;4;97.00;48.00;1978.;20.0;71;Europe
9 Ford Torino 500;19.0;6;250.0;88.00;3302.;15.5;71;US
10 Pontiac Catalina Brougham;14.0;8;400.0;175.0;4464.;11.5;71;US
11 Ford Galaxie 500;14.0;8;351.0;153.0;4154.;13.5;71;US
```

# UN FICHER CSV EST UN FICHER TEXTE À VALEURS SÉPARÉES

**Note : ce fichier CSV possède une ligne d'en-tête**



	A	B	C	D	E	F	G	H	I
1	Car	MPG	Cylinders	displacement	horsepower	Weight	acceleration	Model	Origin
2	Peugeot 504	25.0	4	110.0	87.00	2672.	17.5	70	Europe
3	Audi 100 LS	24.0	4	107.0	90.00	2430.	14.5	70	Europe
4	BMW 2002	26.0	4	121.0	113.0	2234.	12.5	70	Europe
5	Ford F250	10.0	8	360.0	215.0	4615.	14.0	70	US
6	Toyota Corolla	25.0	4	113.0	95.00	2228.	14.0	71	Japan
7	Ford Pinto	25.0	4	98.00	0	2046.	19.0	71	US
8	Volkswagen Super Beetle 117	0	4	97.00	48.00	1978.	20.0	71	Europe
9	Ford Torino 500	19.0	6	250.0	88.00	3302.	15.5	71	US
10	Pontiac Catalina Brougham	14.0	8	400.0	175.0	4464.	11.5	71	US

IL EST PERTINENT DE REPRÉSENTER UN FICHER  
CSV SOUS LA FORME D'UN TABLEAU

Note : ce fichier CSV possède une ligne d'en-tête



# LES DONNÉES D'UN FICHER CSV PEUVENT ÊTRE CHARGÉES DE DEUX MANIÈRES

- DANS UN TABLEAU  
« `string[x, y]` » OU  
« `x` » EST LE NOMBRE DE LIGNE ET  
« `y` » LE NOMBRE DE COLONNE
- DANS UNE COLLECTION  
« `List<List<string>>` » :  
LA PREMIÈRE LISTE ÉNUMÈRE  
LES LIGNES,  
ET LA SECONDE LES COLONNES

AUSSI BIEN POUR **LIRE** QUE  
POUR **ÉCRIRE** UN **FICHER CSV**,  
IL SERA NÉCESSAIRE DE RÉALISER  
**UNE BOUCLE** DANS **UNE BOUCLE**  
(ON PARLE DE BOUCLES **IMBRIQUÉES**)

```
for (int i = 0; i < nombreDeLigne; i++)  
{  
    // Traitement relatif aux lignes  
    for (int j = 0; j < nombreDeColonne; j++)  
    {  
        // Traitement relatif aux colonnes  
    }  
}
```