

C#

LES BASES : ALGORITHMES, VARIABLES, MOTS-CLEFS « IF » ET « FOR »

TOUR D'HORIZON DU C#

- LANGAGE DE PROGRAMMATION DIT « ORIENTÉ OBJET » (MAIS PAS SEULEMENT)
- PROPOSÉ PAR **MICROSOFT** DEPUIS 2002
- TRÈS BIEN INTÉGRÉ À **WINDOWS**
- BASÉ SUR LE **FRAMEWORK .NET** OU SUR LE **FRAMEWORK CORE**
- GRANDE COMMUNAUTÉ DE DÉVELOPPEURS
- APPROCHE MOINS COMPLEXE QUE LE C OU LE C++
- DANS L'ESPRIT : ASSEZ **PROCHE DE JAVA**



LE C# PERMET PRINCIPALEMENT DE FAIRE DES LOGICIELS DE GESTION

- PROGRAMMES EN MODE CONSOLE
(TÂCHES PLANIFIÉES OU SERVICES WINDOWS)
- APPLICATIONS (CLIENT-LOURD) AVEC IHM (WINFORM, WPF)
- SITES WEB (CLIENT LÉGER) TYPE INTRANET, EXTRANET OU INTERNET
(ASPX, RAZOR)

POURQUOI FAIRE UN LOGICIEL DE GESTION ?

Réaliser un traitement de manière rigoureuse

Faire des calculs et des statistiques

Manipuler des fichiers et des données

LES POINTS DE DÉPART DE TOUT LOGICIEL



Un travail sur ordinateur est réalisé à la main et il est rébarbatif ou fastidieux



Un fichier Excel avec des formules ou des macros atteint ses limites : il faut aller plus loin



Un problème organisationnel se pose : un besoin de traitement automatique se fait sentir

À CHAQUE FOIS
QU'UN DÉVELOPPEUR
CONSTRUIT UN LOGICIEL :

IL FAIT COMME LE CUISINIER
QUI VEUT POUVOIR CONCOCTER
UN BON PLAT À COUP SÛR :

IL ÉCRIT UNE RECETTE



UNE BONNE RECETTE DE CUISINE :

Liste les ingrédients nécessaires

Requiert d'avoir des ustensiles
& du matériel à disposition

Énumère les étapes à suivre
pour réussir le plat

EXEMPLE CONCRET DE RECETTE DE CUISINE

Liste les ingrédients nécessaires

Requiert d'avoir des ustensiles et du matériel à disposition

Énumère les étapes à suivre pour réussir le plat

Pâte à crêpes



Ingrediénts	Préparation
300 g de farine	Temps Total : 30 min
3 oeufs entiers	Préparation : 10 min Cuisson : 20 min
3 cuillères à soupe de sucre	
2 cuillères à soupe d'huile	
50 g de beurre fondu	
60 cl de lait	
Ustensiles	
1 louche	
1 fouet	
1 cuillère en bois	
1 Poêle à crêpes	

1 Mettre la farine dans une terrine et former un puits.

2 Y déposer les oeufs entiers, le sucre, l'huile et le beurre.

3 Mélanger délicatement avec un fouet en ajoutant au fur et à mesure le lait. La pâte ainsi obtenue doit avoir une consistance d'un liquide légèrement épais.

4 Faire chauffer une poêle antiadhésive et la huiler très légèrement. Y verser une louche de pâte, la répartir dans la poêle puis attendre qu'elle soit cuite d'un côté avant de la retourner. Cuire ainsi toutes les crêpes à feu doux.



UN BON LOGICIEL DE GESTION :

Liste les variables et tableaux nécessaires

Requiert d'avoir des ressources matérielles et des librairies chargées

Énumère les étapes à suivre pour réussir le traitement

LISTER LES VARIABLES,
ÉNUMÉRER LES ÉTAPES :

C'EST CE QU'ON APPELLE
UN ALGORITHME

SI L'ON EST CAPABLE D'ÉCRIRE
UNE **RECETTE** POUR UN
PLAT QU'ON SAIT FAIRE,

ALORS ON EST CAPABLE
D'ÉCRIRE UN **ALGORITHME** POUR UN
TRAITEMENT QU'ON SAIT RÉALISER

LA DIFFÉRENCE ENTRE UNE RECETTE DE CUISINE ET UN ALGORITHME ?

La première est
exécutée par un **humain**

Le deuxième est exécuté
par un **ordinateur**

LA DIFFÉRENCE ENTRE UNE RECETTE DE CUISINE ET UN ALGORITHME ?

La première est exécutée par un **humain**

C'est intelligent,
ça a de l'intuition,
ça a des souvenirs,
ça possède de l'expérience

Le deuxième est exécuté par un **ordinateur**

C'est vraiment con

LA DIFFÉRENCE ENTRE UNE RECETTE DE CUISINE ET UN ALGORITHME ?

La première est exécutée par un **humain**

Elle peut être approximative

Miser sur le fait que l'humain a déjà quelques connaissances

Qu'il fait preuve de présence d'esprit...

Le deuxième est exécuté par un **ordinateur**

Il doit être très précis

Aller loin dans le détail

Prévoir un maximum de cas de figure

ET SI UN **HUMAIN**
APPLIQUAIT UNE
RECETTE COMME
UN **ORDINATEUR**
APPLIQUE UN
ALGORITHME ?



[HTTPS://WWW.YOUTUBE.COM
/WATCH?v=cDA3_5982H8](https://www.youtube.com/watch?v=cDA3_5982H8)



LES ALGORITHMES, COMMENT ÇA MARCHE ?

- COMME UNE RECETTE, UN ALGORITHME SE LIT « PAS À PAS », LIGNE APRÈS LIGNE
- POUR ÉCRIRE UN ALGORITHME, IL FAUT ARRIVER À « METTRE À PLAT » SON RAISONNEMENT
- IL FAUT ALLER LOIN DANS LE DÉTAIL :
CHAQUE « PROBLÈME » PEUT ÊTRE DÉCOMPOSÉ EN « SOUS-PROBLÈMES »
- BIEN SOUVENT, UN ALGORITHME A DES DONNÉES D'ENTRÉES ET DES DONNÉES DE SORTIE

ESSAYONS D'ÉCRIRE L'ÉTAPE 1 DE CETTE RECETTE SOUS LA FORME D'UN ALGORITHME

GARDEZ CECI À L'ESPRIT :

LECTURE « PAS À PAS »

« METTRE À PLAT » SON RAISONNEMENT

DÉCOMPOSER EN « SOUS-PROBLÈMES »

PRENDRE EN COMPTE LES DONNÉES D'ENTRÉES

Pâte à crêpes



Ingrédients

	300 g de farine
	3 oeufs entiers
	3 cuillères à soupe de sucre
	2 cuillères à soupe d'huile
	50 g de beurre fondu
	60 cl de lait

Ustensiles

	1 louche
	1 fouet
	1 cuillère en bois
	1 Poêle à crêpes

Préparation

Temps Total : 30 min

Préparation : 10 min

Cuisson : 20 min

- 1 Mettre la farine dans une terrine et former un puits.
- 2 Y déposer les oeufs entiers, le sucre, l'huile et le beurre.
- 3 Mélanger délicatement avec un fouet en ajoutant au fur et à mesure le lait. La pâte ainsi obtenue doit avoir une consistance d'un liquide légèrement épais.
- 4 Faire chauffer une poêle antiadhésive et la huiler très légèrement. Y verser une louche de pâte, la répartir dans la poêle puis attendre qu'elle soit cuite d'un côté avant de la retourner. Cuire ainsi toutes les crêpes à feu doux.

VOILÀ CE QUE ÇA POURRAIT DONNER :

1. SE RENDRE DANS LA CUISINE
2. POUR CHAQUE INGRÉDIENT ET POUR CHAQUE USTENSILE :
 - TROUVER LE PLACARD CONTENANT L'INGRÉDIENT OU L'USTENSILE
 - SORTIR L'INGRÉDIENT OU L'USTENSILE
3. PRENDRE UNE BALANCE
4. POSER LA BALANCE SUR LE PLAN DE TRAVAIL
5. ALLUMER LA BALANCE
6. PRENDRE UNE TERRINE
7. POSER LA TERRINE SUR LA BALANCE
8. TARER LA BALANCE
9. PRENDRE LE PAQUET DE FARINE
10. OUVrir le paquet de farine
11. POSITIONNER LE PAQUET DE FARINE AU DESSUS DE LA TERRINE
12. TANT QUE LA BALANCE INDIQUE UNE VALEUR INFÉRIEURE À 300 G :
 - VERSER LA FARINE DANS LA TERRINE
13. ARRÊTER DE VERSER LA FARINE
14. PRENDRE UNE CUILLÈRE
15. FORMER UN PUITS DANS LA FARINE AVEC LA CUILLÈRE

Pâte à crêpes



Ingrediénts

	300 g de farine
	3 oeufs entiers
	3 cuillères à soupe de sucre
	2 cuillères à soupe d'huile
	50 g de beurre fondu
	60 cl de lait

Préparation

Temps Total : 30 min

Préparation : 10 min	Cuisson : 20 min
----------------------	------------------

1 Mettre la farine dans une terrine et former un puits.

2 Y déposer les oeufs entiers, le sucre, l'huile et le beurre.

3 Mélanger délicatement avec un fouet en ajoutant au fur et à mesure le lait. La pâte ainsi obtenue doit avoir une consistance d'un liquide légèrement épais.

4 Faire chauffer une poêle antiadhésive et la huiler très légèrement. Y verser une louche de pâte, la répartir dans la poêle puis attendre qu'elle soit cuite d'un côté avant de la retourner. Cuire ainsi toutes les crêpes à feu doux.

VOICI UN ALGORITHME ÉCRIT EN C#

```
string résultat = "(texte vide)";
string prénom = "Jean-Michel";
DateTime dateDeNaissance = new DateTime(1990, 07, 20);
DateTime uneAutreDate = new DateTime(2010, 08, 22);

if (dateDeNaissance > uneAutreDate)
{
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}

else
{
    for (int i = 0; i < 99; i++)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate)
        {
            résultat = prénom + " avait " + i.ToString() + " an(s) !";
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat);
```

- CET ALGORITHME POSSÈDE TROIS DONNÉES D'ENTRÉES : UN PRÉNOM, UNE DATE DE NAISSANCE ET UNE AUTRE DATE**
- IL POSSÈDE UNE DONNÉE DE SORTIE : LE « RÉSULTAT », QUI EST UNE PHRASE (CHAINE DE CARACTÈRE). CETTE DONNÉE DE SORTIE EST INITIALISÉE DÈS LE DÉBUT.**
- IL CONSISTE À CALCULER UN ÂGE (SI C'EST PERTINENT) ET À DÉTECTOR LE JOUR DE L'ANNIVERSAIRE. IL GÉNÈRE UNE PHRASE EN FRANÇAIS QU'IL AFFICHE À L'UTILISATEUR DANS UNE BOITE DE MESSAGE.**

```

string résultat = "(texte vide)"; // On déclare une variable chaîne contenant "(texte vide)"
string prénom = "Jean-Michel"; // On déclare une variable chaîne contenant "Jean-Michel"
DateTime dateDeNaissance = new DateTime(1990, 07, 20); // On déclare une variable date contenant 20/07/1990
DateTime uneAutreDate = new DateTime(2010, 08, 22); // On déclare une variable date contenant 22/08/2010

if (dateDeNaissance > uneAutreDate) // Si la date de naissance est supérieure à l'autre date
{
    // Oui : la date de naissance EST supérieure à l'autre date !
    // On concatène une réponse pour dire que la personne n'était pas née
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}
else
{
    // Non : la date de naissance N'EST PAS supérieure à l'autre date !
    for (int i = 0; i < 99; i++) // On démarre une boucle, avec une variable i, de type entier (itérateur)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate) // Si la date de naissance + i année(s) est supérieure ou égale à l'autre date
        {
            // Oui : la date de naissance + i année(s) EST supérieure ou égale à l'autre date !
            résultat = prénom + " avait " + i.ToString() + " an(s) !"; // On indique l'âge calculé (qui vaut i)
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat); // On affiche le résultat de l'algorithme dans une boîte de message

```

```
string résultat = "(texte vide)"; // On déclare une variable chaîne contenant "(texte vide)"
string prénom = "Jean-Michel"; // On déclare une variable chaîne contenant "Jean-Michel"
DateTime dateDeNaissance = new DateTime(1990, 07, 20); // On déclare une variable date contenant 20/07/1990
DateTime uneAutreDate = new DateTime(2010, 08, 22); // On déclare une variable date contenant 22/08/2010

if (dateDeNaissance > uneAutreDate) // Si la date de naissance est supérieure à l'autre date
{
    // Oui : la date de naissance EST supérieure à l'autre date !
    // On concatène une réponse pour dire que la personne n'était pas née
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}
else
{
    // Non : la date de naissance N'EST PAS supérieure à l'autre date !
    for (int i = 0; i < 99; i++) // On démarre une boucle, avec une variable i, de type entier (itérateur)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate) // Si la date de naissance + i année(s) est supérieure
        {
            // ou égale à l'autre date !
            résultat = prénom + " avait " + i.ToString() + " an(s) !"; // On indique l'âge calculé (qui vaut i)
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += résultat + " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat); // On affiche le résultat de l'algorithme dans une boîte de message
```

```

string résultat = "(texte vide)"; // Ainsi déclare une variable chaine contenant "texte vide"
string prénom = "Jean-Michel"; // Ainsi déclare une variable chaine contenant "Jean-Michel"
DateTime dateDeNaissance = new DateTime(1990, 07, 20); // Ainsi déclare une variable de type datetime, valeur 1990/07/20
DateTime uneAutreDate = new DateTime(2010, 08, 22); // Ainsi déclare une variable de type datetime, valeur 2010/08/22

if (dateDeNaissance > uneAutreDate) // Si la date de naissance est supérieure à l'autre date
{
    // Alors : la date de naissance est supérieure à l'autre date
    // On complète une chaîne pour afficher le programme d'anniversaire pour nous
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}
else
{
    // Sinon : la date de naissance n'est pas supérieure à l'autre date
    for (int i = 0; i < 99; i++) // On déclare une boucle, avec une variable i, du type entier (entier)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate) // Si la date de naissance + i années est supérieure
        // à une autre date
        {
            // Alors : la date de naissance + i années est égale à l'autre date
            résultat = prénom + " avait " + i.ToString() + " an(s) !"; // On complète l'expression finale par i
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += résultat + " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat); // On affiche finalement un message contenant une boîte de message

```

LE CODE

LES COMMENTAIRES

```
string chainevide = "" // On déclare une variable chaîne contenant "(texte vide)"  
string chaine = "Jean-Michel" // On déclare une variable chaîne contenant "Jean-Michel"  
Date dateNaissance = new Date(1990, 07, 20) // On déclare une variable date contenant 20/07/1990  
Date dateAujourd'hui = new Date(2010, 08, 22) // On déclare une variable date contenant 22/08/2010  
  
if (dateNaissance > dateAujourd'hui) // Si la date de naissance est supérieure à l'autre date  
  
    // Oui : la date de naissance EST supérieure à l'autre date !  
    // On concatène une réponse pour dire que la personne n'était pas née  
    resultat = resultat + "La personne n'a pas été née le " + dateNaissance.toLocaleString()  
  
else  
  
    // Non : la date de naissance N'EST PAS supérieure à l'autre date !  
for (int i = 0; i < 100; i++) // On démarre une boucle, avec une variable i, de type entier (itérateur)  
  
    if (dateNaissance.addYears(i) >= dateAujourd'hui) // Si la date de naissance + i année(s) est supérieure  
        // ou égale à l'autre date  
        // Oui : la date de naissance + i année(s) EST supérieure ou égale à l'autre date !  
        resultat = resultat + " La personne a " + i + " ans en " + dateNaissance.toLocaleString() + " ans en " + dateAujourd'hui.toLocaleString()  
  
        // resultat = resultat + " La personne a " + i + " ans plus le jour de son anniversaire ! "  
  
        break  
  
// On affiche le résultat de l'algorithme dans une boîte de message
```

```
string résultat = "(texte vide)"; // On déclare une variable chaîne contenant "(texte vide)"  
string prénom = "Jean-Michel"; // On déclare une variable chaîne contenant "Jean-Michel"  
DateTime dateDeNaissance = new DateTime(1990, 07, 20); // On déclare une variable date contenant 20/07/1990  
DateTime uneAutreDate = new DateTime(2010, 08, 22); // On déclare une variable date contenant 22/08/2010
```

// (declaration + initialisation) // Ici les deux déclarations sont regroupées la même ligne

- (1) Quel est le type de variable qui contiendra la date de naissance ?
- (2) Si nous devions écrire pour dire que la personne n'était pas née dans un pays, à quelles lignes par rapport à la ligne `DateTime naissance = new DateTime(1990, 07, 20);` devrions-nous ajouter ?

réponse

- (1) Quel est le type de variable utilisée pour stocker la date de naissance ?
`DateTime dateDeNaissance = new DateTime(1990, 07, 20);` // La réponse sera donc : une variable de type `DateTime` (date).

// (declaration+initialisation) // Les deux lignes sont regroupées pour éviter d'écrire deux fois la même chose.

- (2) Quel est le type de variable qui contiendra la date de naissance et pourquoi ?
réponse : Il s'agit d'un `DateTime` car il faut stocker une date et une heure.

réponse : `DateTime dateDeNaissance = new DateTime(1990, 07, 20);` (la date de naissance)

réponse

Problème résolu ! // On affiche le résultat de l'algorithme dans une boîte de message

LA DÉCLARATION DES VARIABLES

DES INSTRUCTIONS CONDITIONNELLES

« IF », « ELSE »

```
using System; // Utilisez cette ligne pour utiliser les instructions conditionnelles
using System.Collections.Generic; // Utilisez cette ligne pour utiliser les listes
using System.Linq; // Utilisez cette ligne pour utiliser les méthodes LINQ
using System.Text; // Utilisez cette ligne pour utiliser les chaînes de caractères
using System.Threading.Tasks; // Utilisez cette ligne pour utiliser les tâches asynchrones
using System.Windows.Forms; // Utilisez cette ligne pour utiliser les formulaires Windows

class Program
{
    static void Main()
    {
        // Définissez une variable de type Date
        DateTime dateDeNaissance = new DateTime(1990, 01, 01);
        DateTime uneAutreDate = new DateTime(1990, 01, 02);

        // Affichez la date de naissance
        Console.WriteLine("La date de naissance est : " + dateDeNaissance.ToString("dd/MM/yyyy"));

        // Affichez l'autre date
        Console.WriteLine("L'autre date est : " + uneAutreDate.ToString("dd/MM/yyyy"));

        // Vérifiez si la date de naissance est supérieure à l'autre date
        if (dateDeNaissance > uneAutreDate) // Si la date de naissance est supérieure à l'autre date
        {
            // Oui : la date de naissance EST supérieure à l'autre date !
            // Pour information : lorsque vous utilisez la fonction ToString(), le type string est automatiquement converti en une chaîne de caractères
            Console.WriteLine("La date de naissance est supérieure à l'autre date !");
        }
        else
        {
            // Non : la date de naissance N'EST PAS supérieure à l'autre date !
            // Pour information : lorsque vous utilisez la fonction ToString(), le type string est automatiquement converti en une chaîne de caractères
            if (dateDeNaissance.AddYears(1) >= uneAutreDate) // Si la date de naissance + 1 année(s) est supérieure ou égale à l'autre date
            {
                // Oui : la date de naissance + 1 année(s) EST supérieure ou égale à l'autre date !
                // Pour information : lorsque vous utilisez la fonction ToString(), le type string est automatiquement converti en une chaîne de caractères
                if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
                {
                    // Vérouillez que la date de naissance + 1 année(s) est toujours plus le jour de son anniversaire !
                }
            }
        }
    }
}
```

Retournez à la page principale | [Accéder au tableau des matières](#) | [Accéder à la liste des cours](#) | [Accéder aux notes de cours](#)

DES AFFECTATIONS DE VARIABLES ET DES CONCATÉNATIONS

```
string prénom = "Jeanne"; // On déclare une variable globale contenant "Jeanne"
string prénom2 = "Jeanne"; // On déclare une variable locale nommée "prénom2"
DateTime dateNaissance = new DateTime(1990, 01, 01); // On déclare une variable globale contenant 1990/01/01
DateTime anniversaire = new DateTime(2014, 06, 20); // On déclare une variable locale nommée 2014/06/20

// On vérifie si une personne n'a pas encore fêté son anniversaire
if (dateNaissance <= anniversaire) { // Si la date de naissance est inférieure à l'anniversaire
    // On concatène une réponse pour dire que la personne n'était pas née
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}

// On vérifie si une personne n'a pas encore fêté son anniversaire
for (int i = 0; i < 10; i++) { // On déclare une boucle avec une variable i, de type entier (entier)
    // On concatène une réponse pour dire que la personne n'a pas encore fêté son anniversaire
    // On vérifie si la date de naissance + i années (i) est inférieure ou égale à l'anniversaire
    if (dateNaissance + i <= anniversaire) { // Si la date de naissance + i années est inférieure ou égale à l'anniversaire
        // On concatène une réponse pour dire que la personne n'a pas encore fêté son anniversaire
        résultat = prénom + " avait " + i.ToString() + " an(s) !"; // On indique l'âge calculé (qui vaut i)
    }
}

rезультат = risultat + " Et c'était pile le jour de son anniversaire !";
```

Résultat : `Mon application` | `Mon application` | `Mon application` | `Mon application` | `Mon application`

UNE BOUCLE « FOR »

```
String titreChaine = "Titre de la chaine"; // On déclare une variable chaîne contenant "Titre de la chaine"
String prenom = "PrenomUtilisateur"; // On déclare une variable chaîne contenant "PrenomUtilisateur"
String nom = "NomUtilisateur"; // On déclare une variable chaîne contenant "NomUtilisateur"
String dateNaissance = "2000-01-01"; // On déclare une variable chaîne contenant "2000-01-01"
String adresse = "123 rue de la paix"; // On déclare une variable chaîne contenant "123 rue de la paix"

// 1. Début de la boucle : Aucun itérateur n'a été créé pour parcourir le tableau liste
// 2. Initialisation : initialisation de l'itérateur et assignation à l'itérateur liste
    i = 0; // Initialisation de l'itérateur i à 0
    // on initialise une chaîne pour dire que la personne n'existe pas encore
    resultat = prenom + " " + nom + " n'existe pas encore." + adresse + ", " + dateNaissance + ", " + adresse;
    resultat += "\n";
    liste.add(resultat);

// 3. Fin : la fin de la boucle (99) correspond à l'index liste
for (int i = 0; i < 99; i++) // On démarre une boucle, avec une variable i, de type entier (itérateur)
{
    // 4. Début de la boucle (i) : i < 99 (i < liste.size()) (i < liste.get(i).length())
        // 5. La boucle de la boucle (i) : liste.get(i) est égale à l'index liste
        resultat[i] = prenom + " " + nom + " " + adresse + ", " + dateNaissance + ", " + adresse;
        // 6. Chaque fois qu'on écrit dans une variable liste, on doit faire une nouvelle écriture dans cette variable
        resultat[i] = resultat[i].replace(" ", "\n");
    }
}
```

Respectez les règles de codage Java pour éviter des erreurs de syntaxe.

L'AFFICHAGE DU RÉSULTAT (LA DONNÉE DE SORTIE)

```
string résultat = "Date anniversaire : ";  
résultat += dateJourAnniversaire.ToString("dd/MM/yyyy");  
résultat += "\n";  
résultat += "Age : ";  
résultat += age.ToString();  
résultat += "\n";  
résultat += "Nombre de mois : ";  
résultat += mois.ToString();  
résultat += "\n";  
résultat += "Nombre d'années : ";  
résultat += années.ToString();  
  
label1.Text = résultat;  
  
// On affiche le résultat dans une boîte de message  
MessageBox.Show(résultat);
```

À SAVOIR POUR CODER EN C# EN TOUTE SÉRÉNITÉ

LE **C#** EST TRÈS « **SUSCEPTIBLE** » : IL FAUT ÊTRE TRÈS **RIGOUREUX**

PRESQUE TOUTES LES LIGNES DE CODE SE **TERMINENT** PAR UN **POINT-VIRGULE « ; »**

LE RESPECT DE LA **CASSE** (MINUSCULES ET MAJUSCULES) EST **TRÈS IMPORTANT**

TOUTE **PARENTHÈSE** OUVERTE DOIT ÊTRE FERMÉE

TOUTE **ACCOLADE** OUVERTE DOIT ÊTRE FERMÉE

VOTRE CODE DOIT RESTER **INDENTÉ**

VOTRE CODE DOIT ÊTRE **COMMENTÉ**

Conseil important :
Lisez les messages d'erreur
Cherchez à les comprendre :
ne les laissez pas vous bloquer

CONCERNANT LA DÉCLARATION DE VARIABLES

Type nomDeLaVariable = ValeurDeLaVariable;

CONCERNANT LA DÉCLARATION DE VARIABLES

```
Type nomDeLaVariable = ValeurDeLaVariable;
```



Chaine de caractères : « string »
Nombre entier : « int »
Nombre à virgule : « decimal »
Booléen : « bool » (pour dire 1 ou 0 ; oui ou non)
Date / date et heure : « DateTime »
Un seul caractère : « char »

Le C# est un
langage
fortement typé

CONCERNANT LA DÉCLARATION DE VARIABLES

Type **nomDeLaVariable** = **ValeurDeLaVariable** ;



Convention, entre nous :

- En français
- Un nom parlant qui décrit le rôle de la variable
- La première lettre en minuscule,
puis chaque première lettre de mot avec une majuscule
- Pas d'espace, pas d'underscore « _ »

CONCERNANT LA DÉCLARATION DE VARIABLES

Type nomDeLaVariable = ValeurDeLaVariable;

Selon le cas :

- Pour les chaînes : entre guillemets double « " »
- Pour les entiers : rien de spécial
- Pour les nombres à virgule : on utilise le point « . » en guise de séparateur suivie de la lettre « m »
- Pour les booléens : « true » pour oui et « false » pour non
- Pour les objets (comme « DateTime ») : on utilise le mot-clé « new »



CONCERNANT LA DÉCLARATION DE VARIABLES

Voici quelques exemples concrets

```
string immatriculation = "CD-458-AB";  
int nombreDePlaces = 5;  
decimal carburantRestant = 15.35m;  
bool estDemaree = true;  
DateTime mec = new DateTime(2008, 4, 20);
```

« *mec* » veut dire « *mise en circulation* »

CONCERNANT LA DÉCLARATION DE VARIABLES

Bon à savoir :

**Pour convertir une variable d'un type de données vers un autre type de données,
il est possible d'utiliser le mot-clef « Convert »
ou les méthodes « Parse() » et « TryParse() » dont disposent certains types**

**Tous les types de variables peuvent être convertis en chaîne de caractère
à l'aide de la méthode « ToString() »**

CONCERNANT L'INSTRUCTION CONDITIONNELLE « IF », « ELSE »

```
if (maVariable == autreVariable)
{
    // Du code
}
else
{
    // Du code
}
```

CONCERNANT L'INSTRUCTION CONDITIONNELLE « IF », « ELSE »

```
if (maVariable == autreVariable)
{
    // Du code
}
else
{
```

- 
- Un « if » a toujours la même forme :
 - Le mot-clef « if »
 - Une parenthèse ouvrante et une parenthèse fermante (qui englobent la « condition »)
 - Une accolade ouvrante, et une accolade fermante (qui englobent le « code »)
 - Le « else » est facultatif

CONCERNANT L'INSTRUCTION CONDITIONNELLE « IF », « ELSE »

```
if (maVariable == autreVariable)
{
    // Du code
}
else
{
    Une condition doit être une valeur booléenne.
Une variable de type « bool » suffit.
} Sinon, il faut comparer deux variables de même type
en utilisant un opérateur de comparaison :
    ▪ « == » pour « est égal à »
    ▪ « != » pour « est différent de »
    ▪ « >= » pour « est supérieur ou égal »
    ▪ « < » pour « est strictement inférieur »
Possible de comparer une variable avec une valeur « en dur »
```



CONCERNANT L'INSTRUCTION CONDITIONNELLE « IF », « ELSE »

```
if (maVariable == autreVariable)
{
    // Du code
}
else
{
    Si plusieurs conditions doivent être vérifiées dans le
    « if », il faut les cumuler avec les opérateurs :
}
    ▪ « && » qui veut dire « and » (« et ») : toutes les
        conditions doivent être vérifiées
    ▪ « || » qui veut dire « or » (« ou ») : seule une
        des conditions doit être vérifiée
```



CONCERNANT L'INSTRUCTION CONDITIONNELLE « IF », « ELSE »

```
if (maVariable == autreVariable)
{
    // Du code
}
else
{
    // Du code
}
```

Bloc de code exécuté si la condition est « vérifiée » (si la condition est vraie)

Bloc de code exécuté si la condition n'est pas « vérifiée » (si la condition est fausse)

Il est possible d'enchaîner plusieurs séquences de « if » en utilisant « else if »

Le mot-clé « switch » combiné à « case » sert également d'instruction conditionnelle

CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```

```
uneVariable = uneVariable + " et voilà !";
```

CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```



```
uneVariable = uneVariable + " et voilà !";
```



L'opérateur « = » est l'opérateur d'affectation.
Il ne doit pas être confondu avec l'opérateur « == » qui est
l'opérateur d'égalité (et qui permet de faire des comparaisons)

CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```



```
uneVariable = uneVariable + " et voilà !";
```

À gauche du « = », la variable que l'on souhaite affecter (que l'on souhaite « alimenter »).
Dans le cadre d'une concaténation, cette variable est nécessairement de type « string ».

CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```

```
uneVariable = uneVariable + " et voilà !";
```

À droite du « = », la valeur que l'on souhaite placer dans la variable.
Il s'agit ici de concaténations (des chaînes mises bout à bout).



CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```

```
uneVariable = uneVariable + " et voilà !";
```

L'opérateur « + » permet de faire des additions et des concaténations.
Utilisé avec des variables de type « string », il fait des concaténations.

CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```

```
uneVariable = uneVariable + " et voilà !";
```

Les parties entre guillemets doubles « " » sont des chaines « en dur » qui sont elles aussi concaténées

CONCERNANT L'AFFECTION DE VARIABLES ET LA CONCATÉNATION

```
uneVariable = uneChaine + " et " + autreChose.ToString();
```

```
uneVariable = uneVariable + " et voilà !";
```



Un appel à la méthode « `ToString()` » est obligatoire pour toutes les variables qui ne sont pas du type « `string` »

CONCERNANT LA BOUCLE « FOR »

```
for (int i = 0; i < 99; i++)  
{  
}  
// Du code
```

Un « **for** » a toujours la même forme :

- Le mot-clef « **for** »
- Une parenthèse ouvrante et une parenthèse fermante (qui contiennent deux point-virgules)
- Une accolade ouvrante, et une accolade fermante (qui englobent le « **code** »)

CONCERNANT LA BOUCLE « FOR »

« i » est une variable de type entier
« 0 » est la valeur de départ de la boucle...

```
for (int i = 0; i < 99; i++)  
{  
    // Du code  
}
```

...et « 99 » est la valeur de fin de la boucle

Ces valeurs, ici exprimées « en dur », peuvent aussi être des variables (préalablement déclarées)

CONCERNANT LA BOUCLE « FOR »

```
for (int i = 0; i < 99; i++)  
{  
    // Du code  
}
```



On parle ici d'incrémentation
(``i`` va augmenter de 1 à chaque itération)

Cette boucle « `for` » va permettre la réalisation
de 99 itérations : de 0 à 98.

À savoir :

- Le mot-clé « `break` » permet de quitter la boucle
- Le mot-clé « `continue` » permet de passer à l'itération suivante

CONCERNANT LA BOUCLE « FOR »

```
for (int i = 0; i < 99; i++)  
{  
    // Du code  
}
```

Il existe d'autres façon de faire des boucles :

- « **foreach ... in** »
- « **while** »
- « **do ... while** »

Vous trouverez de nombreux exemples sur Internet pour savoir vous en servir

[https://openclassrooms.com/fr/courses/1526901-
apprenez-a-developper-en-c/2866811-les-boucles](https://openclassrooms.com/fr/courses/1526901-apprenez-a-developper-en-c/2866811-les-boucles)

Mais savoir faire une boucle « **for** » permet de tout faire

ARRIVEZ-VOUS À MIEUX COMPRENDRE ?

```
string résultat = "(texte vide)";
string prénom = "Jean-Michel";
DateTime dateDeNaissance = new DateTime(1990, 07, 20);
DateTime uneAutreDate = new DateTime(2010, 08, 22);

if (dateDeNaissance > uneAutreDate)
{
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}

else
{
    for (int i = 0; i < 99; i++)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate)
        {
            résultat = prénom + " avait " + i.ToString() + " an(s) !";
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat);
```

ARRIVEZ-VOUS À MIEUX COMPRENDRE ?

```
string résultat = "(texte vide)";
string prénom = "Jean-Michel";
DateTime dateDeNaissance = new DateTime(1990, 07, 20);
DateTime uneAutreDate = new DateTime(2010, 08, 22);

if (dateDeNaissance > uneAutreDate)
{
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}

else
{
    for (int i = 0; i < 99; i++)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate)
        {
            résultat = prénom + " avait " + i.ToString() + " an(s) !";
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat);
```

L'algorithme doit afficher
« Jean-Michel avait 20 an(s) »

CERTAINS ÉLÉMENTS N'ONT PAS ÉTÉ EXPLIQUÉS

```
string résultat = "(texte vide)";
string prénom = "Jean-Michel";
DateTime dateDeNaissance = new DateTime(1990, 07, 20);
DateTime uneAutreDate = new DateTime(2010, 08, 22);

if (dateDeNaissance > uneAutreDate)
{
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";
}

else
{
    for (int i = 0; i < 99; i++)
    {
        if (dateDeNaissance.AddYears(i) >= uneAutreDate)
        {
            résultat = prénom + " avait " + i.ToString() + " an(s) !";
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)
            {
                résultat += " Et c'était pile le jour de son anniversaire !";
            }
            break;
        }
    }
}

MessageBox.Show(résultat);
```

Comment pouvais-je savoir qu'on crée une date ainsi ?

Comment pouvais-je savoir que ceci existe ?

Comment pouvais-je savoir que ceci affiche un message ?

L'ENSEMBLE DES MOTS-CLEFS C# NE PEUT PAS ÊTRE EXPLIQUÉ DANS UN SEUL COURS

```
string résultat = "(texte vide);  
string prénom = "Jean-Michel";  
DateTime dateDeNaissance = new DateTime(1990, 07, 20);  
DateTime uneAutreDate = new DateTime(2010, 08, 22);  
  
if (dateDeNaissance > uneAutreDate)  
{  
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";  
}  
else  
{  
    for (int i = 0; i < 99; i++)  
    {  
        if (dateDeNaissance.AddYears(i) >= uneAutreDate)  
        {  
            résultat = prénom + " avait " + i.ToString() + " an(s) !";  
            if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day  
            {  
                résultat += résultat + " Et c'était pile le jour de son anniversaire !";  
            }  
            break;  
        }  
    }  
}  
  
MessageBox.Show(résultat);
```

Comment pouvais-je
savoir qu'on crée
une date ainsi ?

Ça vient du
Framework

Comment pouvais-je
savoir que ceci existe ?

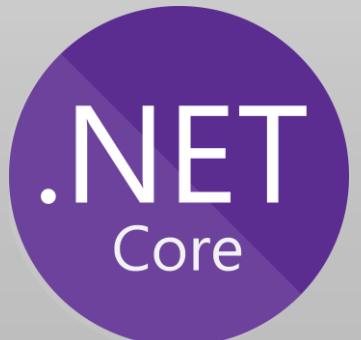
Ça vient du
Framework

Comment pouvais-je savoir
que ceci affiche un message ?

Ça vient du
Framework

MAIS AU FAIT, QU'EST-CE QU'UN FRAMEWORK ?

- C'EST UNE BIBLIOTHÈQUE DE MOTS-CLEFS PRÉCONSTRUITS
- CELA PERMET DE RÉALISER DES ACTIONS COURANTES QUI SONT COMMUNES À DE NOMBREUX LOGICIELS
- C'EST UNE BOITE À OUTILS POUR LES DÉVELOPPEURS QU'ILS UTILISENT AFIN DE NE PAS « RÉINVENTER LA ROUE »
- CHEZ MICROSOFT, IL EN EXISTE DEUX : FRAMEWORK .NET ET FRAMEWORK .NET CORE





POUR MONTER EN COMPÉTENCE IL FAUT **RECHERCHER SUR GOOGLE**

→ VOICI QUELQUES ASTUCES ←



- TOUJOURS COMMENCER AVEC LE MOT « **C#** » (TOUT ATTACHÉ)
- « **C# WINFORM** » SI VOTRE PROBLÈME EST LIÉ À L'IHM (INTERFACE HOMME-MACHINE)
- TOUJOURS S'EXPRIMER EN **ANGLAIS** (MÊME APPROXIMATIF) ; **JAMAIS EN FRANÇAIS**
- OUVRIR PLUSIEURS RÉPONSES (CLIC-MOLETTE) ET LES PARCOURIR RAPIDEMENT,
À LA RECHERCHE DE QUELQUE CHOSE QUI VOUS PARLE, QUI CORRESPONDE À VOTRE BESOIN
- PRIVILÉGIER LES RÉPONSES DU SITE **STACKOVERFLOW** OU DE LA **MSDN** DE MICROSOFT
(MAIS DE NOMBREUX AUTRES SITES PEUVENT APPORTER DES RÉPONSES PERTINENTES)

À force de pratiquer, vous maîtriserez le Framework

VOICI QUELQUES EXEMPLES DE RECHERCHES

```
string résultat = "(texte vide);  
string prénom = "Jean-Michel";  
DateTime dateDeNaissance = new DateTime(1990, 07, 20);  
DateTime uneAutreDate = new DateTime(2010, 08, 22);
```

```
if (dateDeNaissance > uneAutreDate)  
{  
    résultat = prénom + " n'était même pas né(e) le " + uneAutreDate.ToString() + " !";  
}  
else  
{  
    for (int i = 0; i < 99; i++)  
    {  
        if (dateDeNaissance.AddYears(i) >= uneAutreDate)
```

```
{  
    résultat = prénom + " avait " + i.ToString() + " an(s) !";  
    if (dateDeNaissance.Month == uneAutreDate.Month && dateDeNaissance.Day == uneAutreDate.Day)  
    {  
        résultat += résultat + " Et c'était pile le jour de son anniversaire !";  
    }  
    break;  
}
```

```
}  
}  
}  
}  
}  
  
MessageBox.Show(résultat);
```



C# create a specific date



C# add a year to date



C# date get month number



C# winform display message to user



C# get day number from date