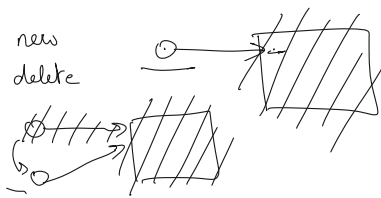
Bonjour tout le monde

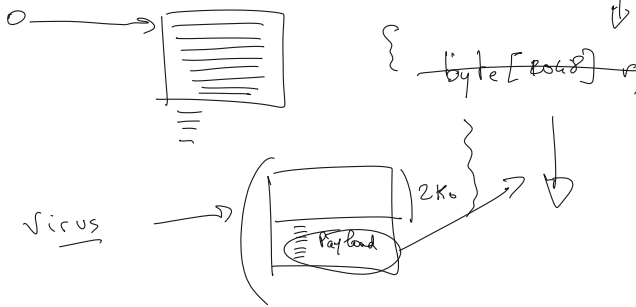
C

malloc
m free

C++

new
delete





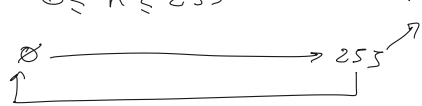
Garbage Collector

- Tapez le même code
- mettez un point d'arrêt (F9)
- Lancez le programme (F5)
- Executez ligne par ligne (F10)

---

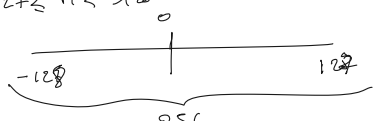
nombre entier non signé 8 bits

$0 \leq n \leq 255$



signé

$-127 \leq n < 128$



8
16
32
64

64 bits valeurs

1
2
4
8
16
32
64
128
256
512
1024

99 valeurs
→ 2 octets –

0 – 9 → 1 octet
0 – 9 → 1 octet

Types de données :

- nombres entiers
- nombres à virgule flottante
- booléen → vrai, faux
- caractère → 1 octet ex: 'O'

(ASCII 1 octet / caractère

Unicode = 8, 16, 32 bits / caractère

| caractères latins
| ≈ 100 caractères différents

- chaîne de caractères (string) ex: "OK"

Catégories de type

Simples

| structures
| champs
| ex: adresse
| date

| Complexes (objets)
| Collections
| - Tableaux
| - Listes

"10"+"5"="105"

10 + 5 = 15

```
// types :
      int,  // entier
        float, double // flottant
        , bool, // booleen
        char, // caractère
        string // chaine de caractères
      /*
       commentaire sur
       plusieur lignes
       */
```

Changements de types

ex: int → float

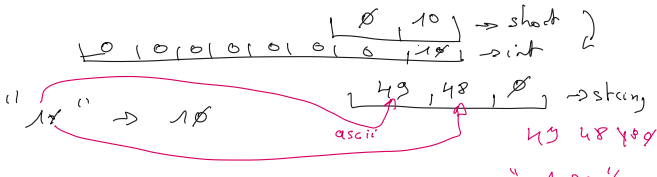
int → string

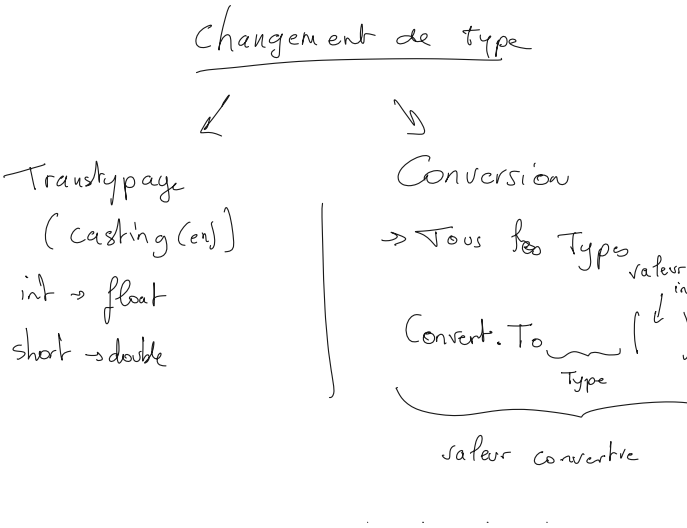
string → int

short ⇔ int

16 bits — 64 bits

short





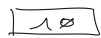

```
// Exercice :
    // 1- Demander nombre au clavier, et l'afficher + 1

    // 2- Demandez 2 nombres au clavier, et affichez leur somme
    Console.Write("Veuillez saisir votre nombre : ");
    string saisie = Console.ReadLine();
    int valeur = Convert.ToInt32(saisie);
    valeur = valeur + 1;    // valeur += 1       valeur++ ( increment )
    Console.WriteLine("Le résultat fait : " + valeur);
```

Conventions d'écriture :

https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/ms229043(v=vs.100)?redirectedfrom=MSDN

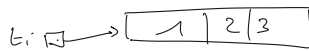
int i; ⇒ 

i = 10 ⇒ 

char c = 'A' = 

string s = "bonjour" 

- s est une référence
- s référence....

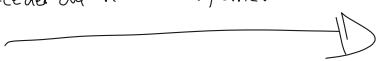


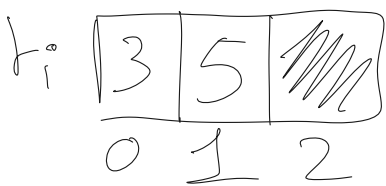
int[] ti = {1, 2, 3}

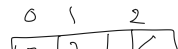
ti[Ø] → n

ti[1] ti[2]

accéder au n ième élément



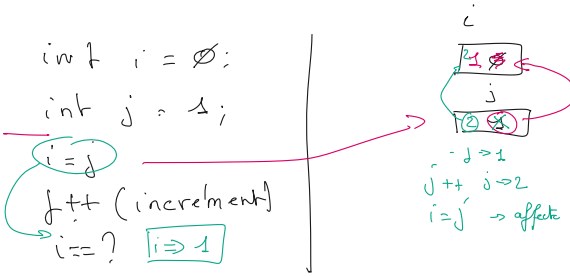


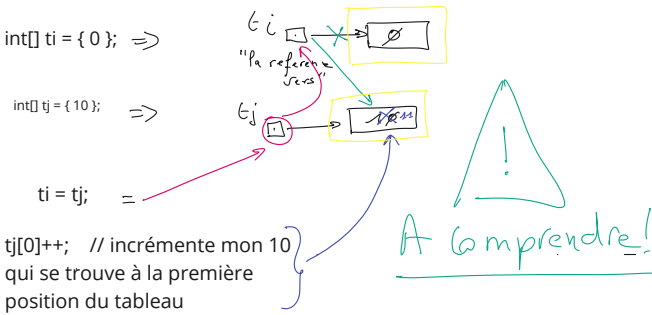


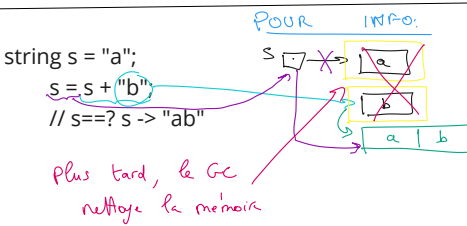
ta = {3, 7, 16}

ta[2] = 16

ta[3] ⇒ erreur


int i = 0;
int j = 1;
i = j
j++ (incrément)
i == ?
i => 1
j
- j -> 1
j++ j -> 2
i = j -> affecte 2 à i


int[] ti = { 0 };
=>
ti
"la référence reste"
int[] tj = { 10 };
=>
tj
ti = tj;
=
tj[0]++; // incrémente mon 10 qui se trouve à la première position du tableau
A comprendre !

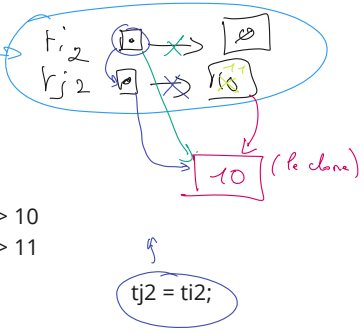

POUR INFO:
string s = "a";
s = s + "b";
// s==? s -> "ab"
s
a
b
a | b
Plus tard, le GC nettoye la mémoire

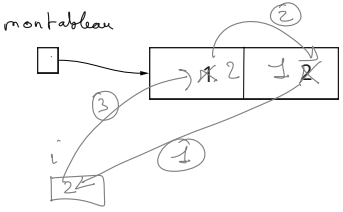
// Si je souhaite obtenir non pas une copie de la référence, mais un DOUBLON du taleau :

```
int[] ti2 = { 0 };
int[] tj2 = { 10 };

ti2 = (int[])( tj2.Clone() );
tj2[0]++;

Console.WriteLine(ti2[0]);  // -> 10
Console.WriteLine(tj2[0]);  // -> 11
```

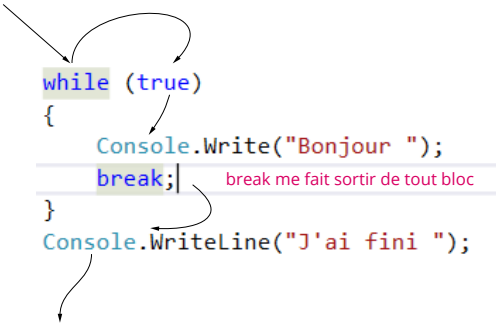




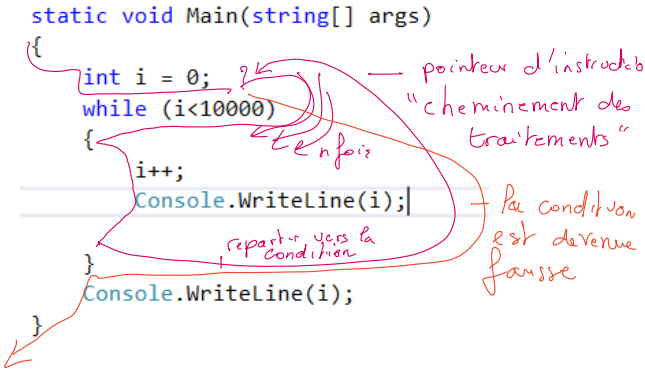

```
// Exercice :
// Faites un tableau avec deux nombres
// intervertissez ces deux nombres
// Envoyez une copie ( un clone ) de votre tableau dans une autre variable
// incrémentez de 1 chaque valeur de ce nouveau tableau.

int[] montableau = { 1, 2 };
// permutation :
i = montableau[0];
montableau[0] = montableau[1];
montableau[1] = i;
// envoi d'un clone
int[] monclonedetableau = (int[])(montableau.Clone());
monclonedetableau[0] = monclonedetableau[0] + 1;    // universel
monclonedetableau[1] += 1;  // j'ajoute ( += ) 1 dans mon tableau
// ou alors j'incrémente ( de 1 ) ma case de mon tableau :
// monclonedetableau[0]++;
```

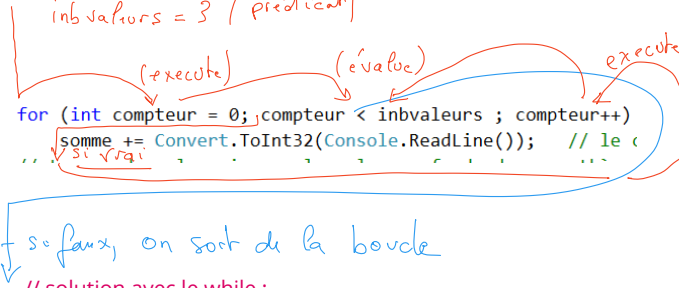
```
// Demandez une valeur à l'utilisateur, et indiquez si cette valeur est supérieure à 10.
    // ex :
    // 5
    // inférieur à 10
    // 12
    // supérieur à 10
    Console.Write("Saisissez votre valeur : ");
    int valeur = Convert.ToInt32( Console.ReadLine());    // n'utilisez pas var
    int valeuracomparer = 10;
    // fausse réponse : ( convertissez en entier )
    if (valeur > valeuracomparer)
       Console.WriteLine("Supérieur a 10");
    else
       Console.WriteLine("Inférieur ou égal a 10");
```



définition d'un "bloc"
est toute série
d'instructions se trouvant
entre { et }



// exercice : affichez les valeurs de 20 a 0




```
// solution avec le while :
        int compteur2 = 0;
        somme = 0;
        while (compteur2++ < inbvaleurs)
            somme += Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("La somme est : " + somme);
```

```
{
    int i = 0;
    while (i<100)
    {
        i++;
        Console.WriteLine(i);
    }
    i = 1;
    int j = i++;
    Console.WriteLine("j="+j);
    Console.WriteLine("i="+i);
    Console.WriteLine("ok");
    i = 1;
    j = i;

    Console.WriteLine("j=" + j);
    Console.WriteLine("i=" + i);
    Console.WriteLine("ok");
    // exercice : affichez les valeurs de 20 a 0
    i = 20;
    //while(i<=20 && i>0) // && et logique, || ( altGr 6 ) : ou logique
    while (i > 0) {
        Console.WriteLine((--i)+1);
    }
    while (i > 0)
    {
        Console.WriteLine((--i) + 1);
    }
    Console.WriteLine("-----------");
```

```
i = 20;
do
{
    Console.WriteLine((--i) + 1);
} while (i > 0);

/*
 postulat que i = 1
 i++ -> je récupère sa valeur, et j'incrémente i    ex : int j = i++;   j vaut 1
 */

// boucle for :
// for ( < initialisation >; < condition >; < iteration > )
// initialisation : non obligatoire, possibilité de fixer des valeurs à des variables.
// condition : ( non obligatoire, vraie par défaut )
//             si la condition est vraie, on entre dans la boucle, sinon on sort ( comme un while )
// itération : non obligatoire, possibilité d'executer un code avant de tester la condition

Console.WriteLine("Via la boucle for : ");
for (int k = 1; k <= 10; k++)
    Console.WriteLine(k);

// boucle infinie :
// for (;;) ;  // équivaut a :for( ;true; );

// exercice :
// Demandez à l'utilisateur combien de nombres il va saisir
// puis demandez ces n nombres, dont vous afficherez la somme à la fin
// ex : combien de nombre ? 3
```

```
// Nombre 1 : 1
// Nombre 2 : 7
// Nombre 3 : 10
// La somme est : 18
Console.Write("Combien de valeurs allez-vous saisir ? ");
string snbvaleurs = Console.ReadLine();
int inbvaleurs = Convert.ToInt32(snbvaleurs);
int somme = 0;
for (int compteur = 0; compteur < inbvaleurs ; compteur++)
    somme += Convert.ToInt32(Console.ReadLine());
Console.WriteLine("La somme est : " + somme);
```

POST FORMATION

Quelques exercices à faire (quand vous avez le temps):

- Mettez des lettres dans un tableau, et triez-le.
- Suggestion : informez-vous de l'algorithme de type "tri à bulles" - compter plusieurs heures...
- Triez (même principe) une chaîne de caractères.
- ex : Bonjour -> Bjnooru
- (Les majuscules seront toujours devant)
- Affichez la factorielle d'une valeur saisie au clavier (ex : 4 -> 1*2*3*4=24) - 15 minutes
- Réalisez le cryptage de César :
- saisissez une chaîne
- demandez la clef (le nombre de décalages)
- affichez la chaine cryptée en retour
- Vous pourrez alors décrypter votre chaine en utilisant une clef négative
- ex truc, valeur de clef : 2
- résultat : vtwe (chaque lettre est décalée de 2)
- vtwe clef -2 -> truc
- Durée estimée : 1 heure.