

The background of the slide is a dark blue image featuring a complex circuit board with gold-colored traces and components. Overlaid on the right side of the circuit is a grid of binary code (0s and 1s) in a light blue/green color, giving it a digital or data-centric appearance.

# Programmation 1: Fondamentaux

Introduction aux tableaux de données

# Qu'est-ce qu'un tableau?

- Un tableau est une variable qui peut contenir plusieurs données d'un même type
  - Il possède donc un identifiant (son nom)
  - il précise le type des données qu'il peut contenir
  - il occupe une taille fixe en mémoire
- Les données du tableau sont de **type de base** ou plus **complexe**
  - entiers, flottants, caractères
  - enregistrements (structures), tableaux (pointeurs)

# Qu'est-ce qu'un tableau?

- Un tableau possède des dimensions
  - un tableau à une dimension est un tableau en ligne.
  - un tableau à deux dimensions possède des lignes et des colonnes
  - un tableau à trois dimensions peut être vu comme un cube
  - il est possible d'avoir encore plus de dimensions, les limites sont fixées par le langage et la mémoire disponible
- Déclaration des tableaux statiques
- `int myTab1[10]` est un tableau à une dimension qui peut contenir dix valeurs entières
- `char myTab2[10][10]` est un tableau de 100 caractères organisé suivant une matrice carrée de 10 lignes et 10 colonnes

# Qu'est-ce qu'un tableau?

- Déclaration des tableaux statiques en pseudocode :

*Déclaration*

*Variable listeNotes en tableau[50] de réels*

*Variable image en tableau[1024][768] d'entiers*

# Qu'est-ce qu'un tableau?

- Taille de la mémoire occupée par un tableau
  - la taille mémoire s'exprime en octets (bytes).
  - les **tableaux statiques** sont des tableaux de **taille fixe** dont l'emplacement en mémoire est réservé une fois pour toute jusqu'à leur destruction.
  - La taille d'un tableau est égale à la taille de l'un de ses éléments multipliée par le nombre total d'éléments qu'il peut contenir
    - En langage C, la **taille d'un type** s'obtient grâce à la fonction **sizeof**
- `int myTab1[10];`  
`int tailleTableauEnOctets = sizeof(int) * 10`
- `char myTab2[10][10];`  
`int tailleTableauEnOctets = sizeof(char) * 10 * 10`

# Accès aux données du tableau

- Chaque élément du tableau est identifié par un indice (index).
- Le premier élément du tableau peut se trouver à l'indice 0 ou à l'indice 1
- En langage C, les tableaux commencent à l'indice 0 et le dernier élément se trouve à l'indice *TailleTableau – 1*
- Il ne faut pas confondre la taille du tableau en octets avec la taille du tableau qui indique le nombre d'éléments qu'il peut contenir.
- Ecriture dans le tableau (pseudocode) :

*MyTab1[indice] <- valeur*

# Accès aux données du tableau

- **Ecriture** dans le tableau à une dimension en langage C :

```
int MyTab[5];    //déclaration
```

```
MyTab1[0]= 10;   // écriture de la valeur 10 à l'indice 0, soit le premier élément
```

```
MyTab1[1]= 20;
```

```
MyTab1[4]= 50;   // écriture de la valeur 50 à l'indice 4, soit le dernier élément
```

On peut initialiser lors de la déclaration, la taille est alors déduite du nombre de valeurs initiales

```
int MyTab[] = {10, 20, 30, 40, 50} ; // déclaration et initialisation d'un tableau de taille 5
```

On peut initialiser un tableau avec une valeur par défaut pour tous les éléments :

```
int MyTab[5] = {0};
```

# Accès aux données du tableau

- **Ecriture** dans le tableau à **Deux** dimensions (C) :

```
int MyTab[3][3]; // Dimension = 3 x 3
```

```
MyTab1[0][0]= 1; MyTab1[0][1]= 2; MyTab1[0][2]= 3;  
MyTab1[1][0]= 4; MyTab1[1][1]= 5; MyTab1[1][2]= 6;  
MyTab1[2][0]= 7; MyTab1[2][1]= 8; MyTab1[2][2]= 9;
```

*MyTab1[indice\_ligne] [indice\_colonne] <- valeur*



# Accès aux données du tableau

- **Lecture** dans le tableau à une dimension en langage C :

```
int MyTab[5];    //déclaration
```

```
MyTab1[0]= 10; // écriture de la valeur 10 à l'indice 0, soit le premier élément
```

```
MyTab1[1]= 20;
```

```
MyTab1[4]= 50; // écriture de la valeur 50 à l'indice 4, soit le dernier élément
```

```
int tmp = MyTab1[0]; // la variable tmp reçoit la valeur à l'indice 0 de MyTab
```

# Accès aux données du tableau

Utilisation de variables pour les indices de tableaux

- les indices de tableaux sont des valeurs entières
- les indices de tableaux ne doivent pas sortir des limites du tableau  
=> *index out of range*

```
int MyTab[10];
```

```
int i = 0;
```

```
MyTab[i] = 35;
```

# Manipulation du tableau

1. Initialiser un tableau de 100 nombres décimaux à la valeur 0.0
2. Demander à l'utilisateur de saisir des valeurs jusqu'à ce qu'il entre une valeur négative et mémoriser le nombre de valeurs saisies
3. Afficher le tableau de valeurs en plaçant 10 valeurs par ligne et avec une précision de 2 chiffres après la virgule
4. Demander à l'utilisateur d'insérer une valeur à la position de son choix
5. Afficher le tableau
6. Demander à l'utilisateur de supprimer une valeur à la position de son choix
7. Afficher le tableau
8. Calculer et afficher la moyenne des valeurs dans le tableau

# Manipulation du tableau

- Initialiser un tableau de 10 lignes et 10 colonnes avec des valeurs entières comprises entre 1 et 20
- Afficher le tableau
- Remplir un second tableau de mêmes dimensions avec les valeurs du premier tableau en créant une symétrie par rapport à une diagonale
- Afficher le deuxième tableau