

# Aide-mémoire

---

## Boucles:

### for

```
for (initialisation; condition; incrémentation) {  
    // instructions à exécuter à chaque itération  
}
```

- **Initialisation** : Cette section initialise une ou plusieurs variables avant que la boucle ne démarre (souvent un compteur).
- **Condition** : Tant que cette condition est vraie, la boucle continue de s'exécuter. Quand la condition devient fausse, la boucle s'arrête.
- **Incrémentation** : À chaque itération, cette étape est exécutée après le bloc de code. Elle permet de modifier la ou les variables du compteur, par exemple en les incrémentant.

### while

```
while (condition) {  
    // instructions à répéter  
}
```

- **condition** : une expression booléenne qui est évaluée avant chaque itération de la boucle.
- Si la condition est vraie (true), le bloc d'instructions est exécuté.
- Si la condition est fausse (false), la boucle s'arrête.

### do...while

```
do {  
    // Instructions à exécuter  
} while (condition);
```

## Switch:

```
char choix;  
...  
  
switch (choix) {  
case '1':  
    option1();  
    break;
```

```
case '2':  
    option2();  
    break;  
case '3':  
    cout << "Au revoir!" << endl;  
    break;  
default:  
    cout << "Choix invalide..." << endl;  
    break;  
}
```

## Fonctions sans paramètres:

sans retour

```
void maFonction1(){  
    //instructions  
}
```

avec retour

```
[type de retour] maFonction2(){  
    //instructions  
    return ... ;  
}
```

## Fonctions avec paramètres:

sans retour

```
void maFonction3([type de param1] nomParam1, [type de param2] nomParam2, ...){  
    //instructions  
}
```

avec retour

```
[type de retour] maFonction4([type de param1] nomParam1, [type de param2]  
nomParam2, ...){  
    //instructions  
    return ... ;  
}
```

avec paramètres (par référence)

```
... maFonction5([type de param1] & nomParam1, [type de param2] & nomParam2, ...){  
    //instructions  
    ...  
}
```

## Valeurs aléatoires

Librairies:

```
#include <cstdlib>  
#include <time.h>    ou    <ctime>
```

Code:

```
srand(time(0)); //Pour utiliser le temps actuel pour générer le seed  
int n = 5; //le nombre max inclus dans le tirage au sort  
int nombreAleatoire = rand() % (n+1);
```