

Pour contrôler plusieurs instructions il faut les regrouper dans un bloc { ... }.

Branchements conditionnels :

```
if (condition1)
    une seule instruction
    ou un bloc
else if(condition2) // optionnel
    une seule instruction
    ou un bloc
else // optionnel
    une seule instruction
    ou un bloc
```

Boucles conditionnelles :

```
while (condition)
    une seule instruction
    ou un bloc

do
    une seule instruction
    ou un bloc
while (condition);
```

Itérations :

```
for (expr_init ; condition ; expr_post)
    une seule instruction
    ou un bloc
```

Les structures de contrôle

```
switch(expr_entière)
{
case cte_entière1: // instruction(s) optionnelles
case cte_entière2:
    une ou plusieurs instructions ; break optionnel
case cte_entière3:
    une ou plusieurs instructions ; break;
...
default: // optionnel
}
```

Pour **switch**, il n'est pas nécessaire d'avoir un bloc associé à un case pour contrôler plusieurs instructions. On peut aussi n'avoir aucune instruction contrôlée.

Branchements inconditionnels:

break quitte une boucle, une itération ou un **switch**
continue ne quitte PAS la boucle mais saute directement en fin du bloc contrôlé

Fin de fonction

return quitte une fonction de type **void**
return valeur quitte la fonction en renvoyant une valeur du même type que la fonction