VARIABLE:

en C, les variables ont nécessairement un type qui doit être précisé à leurs déclarations. Leurs types dépendent de l'utilisation que vous voulez faire de ces variables. Voici des exemples de type principaux.

int: type de base correspondant aux nombres positifs et négatifs
unsigned int: type de variable numérique uniquement positives
char: variable d'un caractère
char*: variable pour stocker des chaînes de caractères
float: variable pour les nombres à virgules.

BOUCLE & CONDITION:
Les boucles et conditions permettent de définir les actions de votre code de manière dynamique. Ce que vous mettrez entre parenthèses "()" seront les conditions à remplir pour réaliser les actions dans les brackets "{}". Voici les principaux.

if() {}: condition qui fera le contenue une seule fois si les conditions dans les parenthèses sont remplies.
while() {}: tant que les conditions dans les parenthèses seront remplies, les actions dans les brackets seront réalisées en boucle.

Les conditions sont les comparaisons:
X == Y: condition qui vérifie si X est identique à Y
X != Y: condition qui vérifie si X est différent de Y
X < Y: condition qui vérifie si X est inférieur à Y
X > Y: condition qui vérifie si X est supérieur à Y
X <= Y: condition qui vérifie si X est inférieur ou égal à Y
X >= Y: condition qui vérifie si X est inférieur ou égal à Y

Il est possible d'ajouter des conditions à l'aide des connecteurs logiques:
&&: les deux conditions doivent être vrai
||: l'une des deux conditions doit être vrai

MAIN:

Tout programme en C à besoin d'une fonction 'main' pour pouvoir se lancer. Une fois compilé, le programme lancera le contenu de la fonction main. Voici un exemple de base.

```
int main(int ac, char **av)
{
	int i = 0;

	while (i < ac) {
		printf(av[i]);
		printf("\n");
```

```
                i = i + 1;
        }
        return (0);
}
```

FONCTION:

Afin de mieux organiser votre code, il est possible de le couper en fonction. Les fonctions peuvent récupérer une liste de variables et renvoie une variable à la fin de leur exécution.

```
int nom_de_fonction(int exemple)
{
        int valeur = 42;

        return valeur;
}
```