

Aide mémoire pour Arduino

La programmation *Arduino* est proche de la programmation C/C++, tout simplement parce que le langage *Arduino* est basé sur du C/C++. Les quelques variations résident majoritairement dans le fait que *Arduino* ajoute ses propres bibliothèques pour le contrôle de son circuit imprimé.

Déclarer une variable

Avant d'utiliser une variable, il faut la déclarer :

```
String nom = "valeur";  
int LED = 13;
```

La valeur est optionnelle et peut être définie plus tard :

```
String test;  
test = "TATA";
```

Il est intéressant de noter qu'*Arduino* a sa propre implémentation des *String*, ce qui peut poser problème en utilisant les *string* implémentés dans C++.

Créer une fonction

La création de fonctions en *Arduino* est pareil qu'en C/C++. Il faut préciser le Type que retourne la fonction (*int*, *String*, *void*...). le type *void* indique que la fonction ne retourne rien :

```
void foo(int parametre){  
    //l'intérieur de votre fonction  
    parametre = 14;  
}
```

La fonction présentée ci-dessus est appelée lorsque l'on exécute :

```
int test = 1;  
foo(test); // test = 14
```

Aspect général d'un code

```
//Declaration de variables globales  
int variable_globale = 1;  
//Initialisation des parametres (Entree/Sortie)  
void setup() {  
    //Faire quelque chose une fois  
    variable_globale = 2;  
}  
//Le code qui sera execute en boucle  
void loop() {  
    //le code qui sera repete  
    variable_globale += 1;  
    variable_globale -= 1;  
}
```

Les spécificités Arduino

Le langage *Arduino* permet d'utiliser plus facilement un circuit imprimé grâce à ses fonctions spécifiques :

Déclaration de l'utilisation d'une broche :

```
int broche = 13;  
pinMode(broche, OUTPUT); // déclarer la broche 13 comme sortie (OUTPUT)
```

Changer le voltage d'une broche :

```
int broche = 13;  
digitalWrite(broche, HIGH); // fournir un courant de 5V dans la broche 13  
digitalWrite(broche, LOW); // le courant passe à 0V dans la broche 13
```

Marquer un temps d'arrêt dans l'exécution du code :

```
delay(1000); // stoppe l'exécution du code pendant 1 seconde
```