

# Les particularités du C / C++ pour l'Arduino

## La fonction `main()` est remplacée par les fonctions `setup()` et `loop()`

### `setup()`

**Description :** La fonction `setup()` est appelée au démarrage du programme. Cette fonction est utilisée pour initialiser les variables, le sens des broches, les bibliothèques utilisées. La fonction `setup` n'est exécutée qu'une seule fois, après chaque mise sous tension ou reset (réinitialisation) de la carte Arduino.

### `loop()`

**Description :** Après avoir créé une fonction `setup()`, qui initialise et fixe les valeurs de démarrage du programme, la fonction `loop ()` (boucle en anglais) fait exactement ce que son nom suggère et s'exécute en boucle sans fin, permettant à votre programme de s'exécuter.

### Exemple :

```
int buttonPin = 3;

void setup() {
  beginSerial(9600);          // la fonction setup initialise la communication série
  pinMode(buttonPin, INPUT); // et une broche utilisée avec un bouton poussoir
}

void loop() { // la fonction loop teste l'état du bouton à chaque passage

  if (digitalRead(buttonPin) == HIGH)
    serialWrite('H');        // et envoie au PC une lettre H si il est appuyé
  else
    serialWrite('L');        // Sinon envoie au PC une lettre L
  delay(1000);
}
```

## les différents types de variables disponibles dans le langage Arduino

Type	Taille en mémoire	Type de données	signe	Valeurs min/max
<b>Valeurs binaires</b>				
<b>boolean</b>	1 bit	Valeur binaire 1/0	non signée	0/1
<b>Valeurs numériques entières signées</b>				
<b>int</b>	2 octets (16 bits)	Valeur entière	signée	-32 768 / +32 767
<b>long</b>	4 octets (32 bits)	Valeur entière	signée	-2 147 483 648 / +2 147 483 647
<b>Valeurs numériques entières non signées</b>				
<b>byte</b>	1 octet (8 bits)	Valeur entière	non signée	0 / +255
<b>unsigned int</b>	2 octets (16 bits)	Valeur entière	non signée	0 / +65535
<b>word</b>	2 octets (16 bits)	Valeur entière	non signée	0 / +65535
<b>unsigned long</b>	4 octets (32 bits)	Valeur entière	non signée	0 / +4 294 967 295
<b>Valeurs numériques à virgule</b>				
<b>float</b>	4 octets (32 bits)	Valeur à virgule	signée	-3.4028235E+38 / +3.4028235E+38
<b>double</b>	4 octets (32 bits)	Valeur à virgule	signée	-3.4028235E+38 / +3.4028235E+38
<b>Caractères</b>				
<b>char</b>	1 octet (8 bits)	Valeur entière - Code ASCII	signée	-128 / +127

## Références Arduino, quelques fonctions utiles...

	Description	Syntaxe	Paramètres	Valeur renvoyée
<b>pinMode()</b>	Configure la broche spécifiée pour qu'elle se comporte soit en entrée, soit en sortie.	pinMode(broche, mode)	<b>broche</b> : le numéro de la broche de la carte Arduino <b>mode</b> : soit INPUT (entrée) ou OUTPUT (sortie)	Aucune
<b>digitalWrite()</b>	Si la broche a été configurée en SORTIE : Met un niveau logique HIGH (HAUT) ou LOW (BAS) sur une broche numérique. Si la broche est configurée en ENTREE : Active (1) ou désactive (0) la résistance interne de "rappel au plus" (pullup).	digitalWrite(broche, valeur)	<b>broche</b> : le numéro de la broche de la carte Arduino <b>valeur</b> : HIGH ou LOW (ou bien 1 ou 0)	Aucune
<b>digitalRead()</b>	Lit l'état (= le niveau logique) d'une broche précise en entrée numérique, et renvoie la valeur HIGH (HAUT en anglais) ou LOW (BAS en anglais).	Val = digitalRead(broche)	<b>broche</b> : le numéro de la broche numérique que vous voulez lire. (int)	Renvoie la valeur HIGH (1) ou LOW (0)
<b>analogRead()</b>	Lit la valeur de la tension présente sur la broche spécifiée. La carte Arduino comporte un convertisseur analogique-numérique 10 bits.	Val = analogRead(br_an)	<b>br_an</b> : le numéro de la broche analogique sur laquelle il faut convertir la tension.	Val : int (0 to 1023) correspondant au résultat de la mesure effectuée
<b>Analog Reference()</b>	Configure la tension de référence utilisée avec les entrées analogiques, 3 options : DEFAULT: tension de référence de 5 V INTERNAL: référence interne de 1.1 V EXTERNAL: tension sur la broche AREF	analogReference(type)	<b>type</b> : le type de référence utilisée soit → DEFAULT, INTERNAL, ou EXTERNAL.	Aucune.
<b>delay(ms)</b>	Réalise une pause dans l'exécution du programme pour la durée (en millisecondes) indiquée en paramètre.	delay (ms)	<b>ms</b> (unsigned long): le nombre de millisecondes que dure la pause .	Aucune
<b>Serial.begin()</b>	Fixe le débit de communication en bits par secondes (l'unité est le baud) pour la communication série.	Serial.begin(9600);	int <b>debit</b> : debit de communication en bits par seconde (ou baud)	Aucune
<b>Serial.print()</b>	Affiche les données sur le port série. Les valeurs de type int, float, <a href="#">byte</a> sont affichés sous la forme de caractères ASCII. Les caractères et les chaînes sont affichés tels que.	<u>SERIAL.PRINT(VAL)</u> Serial.print(val, format)	<b>val</b> : la valeur à afficher. N'importe quel type de données. <b>format</b> : spécifie la base utilisée (pour les nombres entiers) ou le nombre de décimales (pour les nombres de type float)	Aucune
<b>Serial.println()</b>	Idem Serial.print(), puis rajoute un saut de ligne .	<u>SERIAL.PRINTLN(VAL)</u> Serial.println(val, format)	Idem Serial.print()	Aucune

