LANGAGE ARDUINO 1/2

Les parties indiquées ainsi sont à remplacer par les valeurs correspondantes à votre programme.
Broche : à remplacer SOIT par le numéro de la broche SOIT par le nom donné à la broche avant le setup.
Attention : Points virgules à la fin de chaque instruction. Ne pas oublier les parenthèses et les accolades. Utiliser uniquement les caractères alphanumériques.

2 slashs avant	`// cette ligne est un commentaire sur UNE SEULE ligne`	Commentaire sur **UNE SEULE ligne**
1 slash + 1 astérisque avant ET 1 astérisque + 1 slash après	`/*cette ligne est un commentaire sur PLUSIEURS lignes qui sera ignorée par le programme, mais pas par celui qui lit le code */`	Commentaire sur **PLUSIEURS lignes**

Avant le setup :

const int *nom_variable*=*broche* ;	Donner un nom à une broche
int *nom_variable*=*valeur* ;	Créer une variable 16 bits de type entier et lui donner une valeur initiale
long *nom_variable*=*valeur* ;	Créer une variable 32 bits de type entier et lui donner une valeur initiale
char *nom_variable*=*valeur* ;	Créer une variable 8 bits de type entier et lui donner une valeur initiale
float *nom_variable*=*valeur* ;	Créer une variable 32 bits de type entier et lui donner une valeur initiale
boolean *nom_variable* =*valeur* ;	Créer une variable booléenne et lui donner une valeur initiale

setup() :

pinMode(*broche*, OUTPUT) ;	Broche utilisée en sortie (actionneur)
pinMode(*broche*, INPUT) ;	Broche utilisée en entrée (capteur)

loop() (et setup()) :

nom_variable=digitalRead(*broche*) ;	**Lire capteur numérique**
nom_variable=analogRead(*broche*) ;	**Lire capteur analogique**

digitalWrite(*broche*, HIGH) ;	Mettre la sortie à l'état **haut**
digitalWrite(*broche*, LOW) ;	Mettre la sortie à l'état **bas**

delay (*durée en millisecondes*) ;	**Pause** en millisecondes (= 10^{-3} secondes)
delayMicroseconds (*durée en microsecondes*) ;	**Pause** en microsecondes (= 10^{-6} secondes)

while (*condition à valider pour exécution de la boucle jusqu'à ce qu'elle devienne fausse*) {
instructions à exécuter si la condition est vraie avant de l'exécuter ;
}
	Boucle while

do { *instructions à exécuter* ; } while(*condition à tester après chaque exécution de la boucle pour la relancer ou non*);	**Boucle do...while**

for (int *variable* ; *condition à tester avec la variable* ; *incrémentation de la variable*) {
instructions à exécuter tant que la condition est vraie ;
}
	Boucle for

while (1) {
instructions à exécuter ;
}
	Boucle infinie

void *NomDeLaFonction* (char *nom_variable*) {
instructions à exécuter ;
}
	Créer une fonction avec/sans paramètre, sans valeur de retour

NomDeLaFonction (*Valeur du ou des paramètre(s) à intégrer pour cette lecture*);	**Appeler une fonction avec/sans paramètre,** sans valeur de retour

Les parties indiquées ainsi sont à remplacer par les valeurs correspondantes à votre programme.
***Broche** : à remplacer SOIT par le numéro de la broche SOIT par le nom donné à la broche avant le setup.*

Conditions :

if (*condition à tester*) {
instructions à exécuter si la condition est vraie ;
}
	if (si)

if (*condition à tester*) {
instructions à exécuter si la condition est vraie ;
}
else if (*condition à tester*) {
«instructions à exécuter si la condition est vraie» ;
}
	else if (sinon si)

if (*condition à tester*) {
instructions à exécuter si la condition est vraie» ;
}
else {
instructions à exécuter si la condition est fausse ;
}
	else (sinon)

Signes opératoires

Symboles	Signification
==	...est égale à...
<	...est inférieur à...
<=	...est inférieur ou égale à...
>	...est supérieur à...
>=	...est supérieur ou égale à...
!=	...est différent de...

Opérateurs logiques :

variable && *variable*	...et...	(=and)
variable || *variable*	...ou...	(=or)
!*variable*	inverse de...	(=not)

Opérations :

variable + *variable*	Additions
variable - *variable*	Soustractions
variable * *variable*	Multiplications
variable / *variable*	Divisions
variable++	Incrémentation
variable--	Décrémentation

Bluetooth :

#include <SoftwareSerial.h>	Inclure la bibliothèque « SoftwareSerial » *
SoftwareSerial bluetooth (Rx, Tx) ;	Nouveau port série émulé (exmple : bluetooth) avec ports Rx et Tx (remplacer Rx et Tx par numéro de broche) *
bluetooth.begin(9600) ;	Initialisation port « bluetooth » (vitesse standard : 9600 bauds) **

if (bluetooth.available()) {
variable = bluetooth.read() ;
}
	Lorsque le module bluetooth donne des informations, les enregistrer dans une *variable* **

* Ne pas utiliser ces instructions si le module Bluetooth est branché sur les broches Rx et Tx natives de l'Arduino (0 et 1)
** Si le module Bluetooth est branché sur les broches Rx et Tx natives d'un Léonardo remplacer bluetooth par Serial1

Port série (ordinateur) :

Serial.begin(9600) ;	Initialisation port série (vitesse standard : 9600)
Serial.print("texte à envoyer") ;	Message **sans** saut de ligne
Serial.println("texte à envoyer") ;	Message **avec** saut de ligne
Serial.print(*variable*) ;	Valeur variable **sans** saut de ligne
Serial.println(*variable*) ;	Valeur variable **avec** saut de ligne