

## TIPI DI DATI (utilizzabili negli sketch Arduino)

Sketch Arduino (derivato da C/C++)	Descrizione dato	Dimensione
<b>bool</b>	può assumere solo due valori: <b>true</b> o <b>false</b>	<b>1 bit</b>
<b>char</b>	numeri o caratteri (alfanumerico)	<b>1 B (8 bit)</b>
<b>String</b>	Parole formate da uno o più caratteri	
<b>byte</b>	numeri interi da 0 a 255	<b>1 B (8 bit)</b>
<b>short</b>	numeri interi da -32.768 a 32.767	<b>2 B</b>
<b>int</b>	numeri interi da -32.768 a 32.767 (ATmega based boards)	<b>2 B</b>
<b>unsigned int</b>	numeri interi da 0 a 65.535	<b>2 B</b>
<b>word</b>	numeri interi da 0 a 65.535	<b>2 B</b>
<b>long</b>	numeri interi da -2.147.483.648 a 2.147.483.647	<b>4 B</b>
<b>unsigned long</b>	numeri interi da 0 a 4.294.967.295	<b>4 B</b>
<b>float</b>	numeri decimali da $-3,4 \cdot 10^{38}$ a $3,4 \cdot 10^{38}$ (circa)	<b>4 B</b>
<b>void</b>	<b>nessun</b> valore	<b>-</b>

## DICHIARAZIONI DI VARIABILI

Una variabile è un contenitore utile per nominare e memorizzare un valore (numerico o alfanumerico) da utilizzare all'interno di un programma.

Come suggerisce il loro nome, le variabili assumono valori che possono essere cambiati all'interno del programma.

Le "costanti" il cui valore non cambia mai, sono un caso particolare delle variabili.

Le variabili, prima del loro utilizzo, devono essere dichiarate (cioè definite).

In particolare bisogna associare il nome della variabile al tipo di dato.

Per dichiarare una costante bisogna anteporre la parola chiave "**const**"

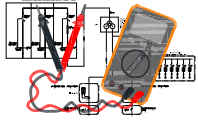
Esempi:

**int** led; (si definisce una variabile di nome "led" che può assumere un valore di tipo "int")

**char** lettera; (si definisce una variabile di nome "lettera" che può assumere un valore di tipo "char")

**bool** flag; (si definisce una variabile di nome "flag" che può assumere un valore di tipo "bool")

**const float** piGreco; (si definisce una costante di nome "piGreco" che può assumere un valore di tipo "float")



Un programma può avere due o più funzioni.

Una funzione si distingue da una variabile per la presenza delle parentesi tonde.

Il tipo di dato legato al nome della funzione esprime la tipologia di valore che una funzione può restituire.

Devono essere sempre presenti le funzioni:

**void setup()** (contiene il codice da eseguire solo una volta, all'avvio del programma)

**void loop()** (contiene il codice da eseguire all'infinito)

Una variabile dichiarata all'interno di una funzione può essere utilizzata solo all'interno della stessa funzione.

Una variabile dichiarata all'esterno delle funzione può essere utilizzata in tutte le funzioni.

## ASSEGNAZIONE DI UN VALORE ALLA VARIABILE

Per assegnare un valore ad una variabile già dichiarata:

**led=45;** (si assegna il valore 45 alla variabile led)

**lettera='h';** (si assegna il valore "h" alla variabile lettera)

**flag=true;** (si assegna il valore true alla variabile flag)

Si può dichiarare la variabile e assegnarne il valore in un'unica istruzione:

**int led = 7;**

**word pippo = 7;**

**const float piGreco = 3.14;**

**N.B. - La dichiarazione di una variabile viene fatta una sola volta, l'assegnazione può essere fatta infinite volte, per modificare il valore della variabile.**

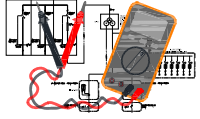
## VISIBILITA' DELLE VARIABILI

La posizione in cui viene dichiarata la variabile determina la sua visibilità all'interno del programma e la capacità, da parte di alcune parti del programma, di utilizzarla.

Una variabile **globale** è una variabile che può essere vista e utilizzata in ogni punto del programma, dentro o fuori ogni funzione del programma.

Una variabile per essere globale deve essere dichiarata all'esterno delle funzioni, solitamente si dichiara una variabile globale all'inizio del programma, prima della funzione setup().

Una variabile è detta **locale** quando è definita all'interno di una funzione, essa è visibile e può essere utilizzata solo all'interno della funzione in cui è stata dichiarata. E' quindi possibile avere due o più variabili con lo stesso nome (cosa sconsigliata per evitare confusione, ma comunque possibile) se presenti all'interno di funzioni diverse.



**Caso particolare:** una variabile **locale** può essere definita all'interno di un ciclo (ad esempio il ciclo for) e pertanto visibile solamente dentro quel ciclo.

Di seguito un esempio:

```
int varGlobale;           // 'varGlobale' è visibile da ogni funzione
void setup() {
    int varSetup = 12;    // 'varSetup' può essere utilizzata solo dentro la funzione
                           setup
    varGlobale = 0;       // 'varGlobale' assume il valore "0" durante lo start-up
}

void loop() {
    int varLoop = 12;     // 'varLoop' può essere utilizzata solo dentro la funzione loop
    for (int i=0; i<20; i++){ // 'i' è visibile solamente all'interno del ciclo for
        varLoop= varLoop+i; // 'varLoop' viene sommato il valore di "i"
        varGlobale=i; }    // 'varGlobale' assume il valore della variabile "i"
    varLoop= varGlobale+100;
}
```