

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

**La comunicazione professionale al momento giusto**

*Linguaggio C: lezione 1*

*Di*

*Andrea Barbieri*

*Lezione e video tutorial*

BTF Traduzioni SEO Sviluppo Web

[btfraduzioniseoweb@gmail.com](mailto:btfraduzioniseoweb@gmail.com)

What's up: +393296664935

Piazza Giuseppe Mazzini, 46

53036 Poggibonsi (SI), ITALY

P.IVA: 01407660529

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## **La comunicazione professionale al momento giusto**

### Sommario

<b>Introduzione</b>	<b>3</b>
<b>La memoria Ram del Computer</b>	<b>3</b>
<b>Domande e risposte</b>	<b>4</b>
<b>Tabella delle funzioni di escape più comuni</b>	<b>11</b>
<b>Parole chiave del linguaggio C</b>	<b>13</b>
<b>Variabili in C</b>	<b>14</b>
<b>Dichiarazione delle variabili</b>	<b>15</b>
<b>Inizializzazione delle variabili</b>	<b>17</b>
<b>Operatore sizeof()</b>	<b>18</b>

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

**La comunicazione professionale al momento giusto**

## **Introduzione**

Nella lezione non è stato possibile inserire tutto il materiale che avrei voluto. Quindi ho preparato questo breve report per inserire anche altre nozioni che saranno utili in futuro.

Ho inserito anche dei sorgenti. Digitali e compilati. Ti consiglio di vedere [lezione e video tutorial](#).

## **La memoria RAM del computer**

Tutti i computer usano la memoria ad accesso casuale (RAM, Random Access Memory) per manipolare i dati che stanno elaborando in un dato momento. Ogni computer ha un determinato quantitativo di RAM installata. **La**

**memoria RAM si definisce anche memoria volatile.**

Perché viene ricreata ogni volta che si spegne e si riaccende il computer. La dimensione della Memoria Ram si misura in megabyte (MB). Un megabyte di memoria equivale a

1024 kilobyte e 1 kilobyte di memoria equivale a 1024 byte.

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## **La comunicazione professionale al momento giusto**

Un sistema con 4 MB di memoria equivale a  $4 * 1024$  (MB) \* 1024 (KB) = 4194304 byte di RAM.

Il byte è l'unità di misura di tutte le memorie informatiche. La memoria RAM è strutturata in modo sequenziale. Ogni byte è messo di seguito all'altro. Ogni byte ha un indirizzo unico che lo identifica. Gli indirizzi sono assegnati alle locazioni di memoria in modo progressivo, partendo dall'indirizzo 0. La memoria RAM ha diversi utilizzi. Ma l'unico che interessa il programmatore è l'**archiviazione dei dati**.

### **Domande e risposte corso C**

**Se voglia passare ad un amico un programma in C, quali file si devono copiare ?**

*Una delle caratteristiche positive del C è di essere un linguaggio compilato.*

*Questo significa che dopo aver fatto la compilazione del sorgente si ha un programma eseguibile autosufficiente. Se*

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## **La comunicazione professionale al momento giusto**

*hai intenzione di passare il programma `ciao.c` a chi si occupa di programmazione C si può farlo senza problemi. E' necessario trasferire il programma eseguibile `ciao.exe`. Non serve il codice sorgente `ciao.c` nel il file oggetto `ciao.obj`. Inoltre, non serve neanche il compilatore C.*

***Dopo aver sviluppato un programma eseguibile è importante conservare il file sorgente (.c) o il file oggetto (.obj) ?***

*Se si cancella il file sorgente non sarà più possibile apportare nuove modifiche al programma in futuro. Il file sorgente deve essere conservato. Per il file oggetto ci sono*

# BTF Traduzioni SEO Sviluppo Web

Traduzioni Siti Web Seo Marketing Battitura Testi

## La comunicazione professionale al momento giusto

*delle motivazioni per conservarli. Ma queste motivazioni sono per una conoscenza del C più approfondita. Per il momento di possono cancellare senza problemi. Per ottenerli di nuovo è sufficiente compilare nuovamente il programma attraverso il file sorgente. Di solito gli ambienti di sviluppo (come per esempio CodeBloks, Eclipse, Anjuta ed altri) creano anche altri file oltre il sorgente (.c), l'oggetto (.obj) e all'eseguibile (.exe). Fino a quando hai il sorgente (.c) si può sempre generare il file secondari.*

***Se il compilatore ha integrato un editor di testo siamo obbligati ad usarlo ?***

*Assolutamente no. Si può usare un altro tipo di editor. L'unica condizione da rispettare che salvi il codice sorgente in formato di testo puro. Se il compilatore viene fornito con un editor di testo provatelo. Se non ti trovi bene puoi usare altre soluzioni. Gli editor integrati nei compilatori migliorano di giorno in giorno. Alcuni formattano automaticamente il sorgente in codice C. Altri tipi colorano in maniera diversa le varie parti del codice*

# BTF Traduzioni SEO Sviluppo Web

Traduzioni Siti Web Seo Marketing Battitura Testi

## La comunicazione professionale al momento giusto

*sorgente. Per rendere più semplice l'identificazione degli errori.*

***E' consigliabile ignorare i messaggi di avvertimento (warning) del compilatore ?***

*Alcuni messaggi di avvertimento non portano conseguenze al funzionamento del programma. Ma questo non è sempre vero. Infatti, se il compilatore fa notare dei messaggi di errore o avvertimenti significa che nel programma c'è qualcosa che non va. Ci sono compilatori che consentono di settare il livello di pericolo che fa apparire la segnalazione. In ogni programma è sempre meglio verificare ogni messaggio di avvertimento. E cercare una soluzione invece di andare avanti. Il risultato migliore è di sicuro un programma senza alcun tipo di errore o avvertimento. Anche un solo errore ferma il procedimento di compilazione.*

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## **La comunicazione professionale al momento giusto**

### **Quale effetto hanno i commenti sulla compilazione dei programmi ?**

*I commenti hanno il solo scopo di essere utili al programmatore. Quando il compilatore esegue la conversione da codice sorgente a codice oggetto elimina ogni commento. Questo avviene perché i commenti non influiscono sul comportamento del programma eseguibile.*

*Un programma con tanti commenti ha la stessa velocità d'esecuzione di uno che ne contiene pochi o punti. I commenti allungano il codice sorgente. Questo è poco importante perché permettono di rendere più leggibile in codice sorgente soprattutto quando è necessario aggiornarlo.*

### **Qual è la differenza tra un'istruzione e un blocco ?**

*Un blocco è un insieme di istruzioni racchiuse tra parentesi graffe {}. Il blocco può essere utilizzato ovunque si possibile usare un'istruzione.*



# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## **La comunicazione professionale al momento giusto**

**In quale modo si può ottenere un elenco delle funzioni disponibili ?**

*La maggior parte di compilatori ha un manuale istruzioni che si occupa di dare informazioni sulle funzioni di libreria presenti. Di solito sono riportate in ordine alfabetico.*

*Oppure puoi acquistare un manuale di programmazione C in cui trovi tutte le funzioni ANSI e non.*

Adesso vediamo nuovi programmi C con cui esercitarsi. Ti consiglio di digitarli nell'editor con calma e attenzione.

Nessun manuale è utile se non si svolge anche un'attività di programmazione. Sono programmi facili per te che inizio sono un a scoglio enorme da superare.

In questo programma e nel seguente si stampa una stringa di testo nello schermo. La stringa di testo è stampata in modo diverso nel secondo programma.

Nel secondo programma viene usata la funzione di escape /n che ha il compito di andare a capo.

# BTF

## Traduzioni

# SEO Sviluppo Web

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## La comunicazione professionale al momento giusto

```
1
2 /* Il programma scrive su schermo la frase "Tre casettine dai tetti aguzzi" */
3
4 #include <stdio.h>
5
6 main()
7 {
8     printf("Tre");
9     printf(" casettine");
10    printf(" dai");
11    printf(" tetti");
12    printf(" aguzzi");
13
14
15 }
```

---

```
1
2 /* Simile al precedente programma ma con l'aggiunta di una funzione di escape */
3 #include <stdio.h>
4
5 main()
6 {
7     printf("Tre\n");
8     printf(" casettine\n");
9     printf(" dai\n");
10    printf(" tetti\n");
11    printf(" aguzzi\n");
12
13
14 }
```

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

**La comunicazione professionale al momento giusto**

**Tabella delle funzioni di escape più comuni:**

`\n` vai a capo

`\t` salta una tabulazione

`\b` ritorna un carattere indietro (backspace)

`\a` suona il campanello del computer

`\\` stampa il carattere `\`

`\”` stampa il carattere “

In questo altro programma si calcola l'area del rettangolo.

# BTF

## Traduzioni

# SEO Sviluppo Web

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## La comunicazione professionale al momento giusto

```

1
2
3 /* Calcolo area rettangolo */
4 #include <stdio.h>
5
6 main()
7 {
8     int base;
9     int altezza;
10    int area;
11
12    base = 3;
13    altezza = 7;
14
15    area = base * altezza;
16
17    printf("%d\n", area);
18
19 }

```

Abbiamo creato delle variabili di tipo int e poi le abbiamo inizializzate tramite l'operatore di assegnamento = .

In questo programma abbiamo usato l'indicatore di conversione %d.

Vediamo la tabella completa

Indicatore	Significato	Tipi convertiti
%c	Carattere singolo	char
%d	Intero decimale con segno	int, short

%ld	Intero long decimale con segno	long
%f	Decimale in virgola mobile	float, double
%s	Stringhe di caratteri	Array di char
%u	Interi decimali senza segno	unsigned int, unsigned char
%lu	Interi decimali long senza segno	unsigned long

## Parole chiave del linguaggio C

Queste sono le parole chiavi del linguaggio C e non devono essere usate come nomi di variabili e costante. Il compilatore genererà un messaggio di errore.

auto	int	extern	union
break	long	float	unsigned
case	register	for	void
char	return	goto	volatile
const	short	if	while
continue	signed		
default	sizeof		
do	static		

# BTF Traduzioni SEO Sviluppo Web

Traduzioni Siti Web Seo Marketing Battitura Testi

## La comunicazione professionale al momento giusto

double	struct
else	switch
enum	typedef

### Varabili in C

Si può definire *variabile* un'etichetta associata ad una locazione di memoria. Usare il nome della variabile significa riferirsi ai dati archiviati in quella posizione.

Analizziamo come sono composti i nomi delle variabili in linguaggio C.

Per creare una variabile si devono seguire delle regole proprie del linguaggio di programmazione:

- Il nome può contenere lettere, numeri e il trattino di sottolineatura \_
- Il primo carattere del nome deve essere sempre una lettera. Si può usare anche il trattino di sottolineatura come primo carattere. Ma non è consigliato usarlo.

# **BTF** **Traduzioni** **SEO Sviluppo Web**

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## **La comunicazione professionale al momento giusto**

- In linguaggio C le lettere sono case insensitive. Questo significa che c'è differenza tra le lettere maiuscole e le lettere minuscole. La variabile gatto è diversa dalla variabile Gatto. Il compilatore le tratta come due entità diverse.
- Le parole chiave del C non possono essere usate come nomi di variabili. Abbiamo visto le parole chiave del C nel precedente paragrafo.

### **La dichiarazione delle variabili**

Per poter usare le variabili è necessario prima dichiararle. La dichiarazione fa sapere al compilatore il nome, il tipo di variabile e opzionalmente il valore di inizializzazione della variabile. Se si tenta di usare una variabile che non è stata dichiarata il compilatore genera un messaggio d'errore.

Questo è il formato per la dichiarazione di una variabile:

# BTF Traduzioni SEO Sviluppo Web

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## La comunicazione professionale al momento giusto

*nometipo nomevariabile;*

*nometipo*: specifica il tipo di variabile. E deve essere inserita una parola chiave del C.

*nomevariabile*: è il nome della variabile. E deve rispettare le regole già spiegate nel precedente paragrafo.

E' possibile dichiarare più di una variabile dello stesso tipo in una riga:

*int conta, numero, inizio; /\* tre variabili intere \*/*

*float percento, totale; /\* due variabili a virgola mobile \*/*

Dagli esempi fatti si nota bene che non è necessario ripetere il tipo di variabile. Ma si scrive solo all'inizio.

La posizione del codice in cui è fatta la dichiarazione delle variabili è importante. Più avanti conosceremo meglio le variabili globali e locali. Nei codici proposti ho usato solo variabili globali.



# BTF Traduzioni SEO Sviluppo Web

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## La comunicazione professionale al momento giusto

Le variabili globali hanno le caratteristiche di avere influenza su ogni parte del programma.

### Inizializzazione delle variabili

Quando si dichiara una variabile si chiede al compilatore di riservare dello spazio in memoria per la variabile. Quando si dichiara una variabile il suo valore è imprevedibile. E' sempre meglio inizializzare una variabile con un oggetto. Un'espressione o più comunemente un numero. Per evitare che vada a sovrascrivere dei dati presenti nel computer. Portando al blocco totale della macchina.

Per inizializzare una variabile si usa l'operatore di assegnamento = .

```
int conta; /* riserva dello spazio in memoria per conta */
```

```
conta = 0; /* memorizza 0 in conta */
```

*E' importante ricordare di non inizializzare il tipo di variabile al di fuori dell'intervallo previsto.*

# BTF

## Traduzioni

# SEO Sviluppo Web

*Traduzioni Siti Web Seo Marketing Battitura Testi*

## La comunicazione professionale al momento giusto

### L'operatore sizeof()

L'operatore sizeof da come risultato la dimensione di un oggetto. In questo programma da in byte la dimensione di ogni tipo di variabile. Il risultato può variare da computer a computer. Prova a digitarlo e compilarlo.

```
1
2 /* sizeof.c - Programma che indica la dimensione dei tipi */
3 /* di variabili C in byte */
4
5 #include <stdio.h>
6
7 int main()
8 {
9     printf("\nUn char           è %d byte", sizeof(char));
10    printf("\nUn int            è %d byte", sizeof(int));
11    printf("\nUn short           è %d byte", sizeof(short));
12    printf("\nUn long            è %d byte", sizeof(long));
13    printf("\nUn unsigned char     è %d byte", sizeof(unsigned char));
14    printf("\nUn unsigned int      è %d byte", sizeof(unsigned int));
15    printf("\nUn unsigned short    è %d byte", sizeof(unsigned short));
16    printf("\nUn unsigned long     è %d byte", sizeof(unsigned long));
17    printf("\nUn float            è %d byte", sizeof(float));
18    printf("\nUn double           è %d byte", sizeof(double));
19
20    return 0;
21
22
23
24
25
26 }
27
```