

## ANSI C – Array

Nono tutorial dedicato all'apprendimento del linguaggio di programmazione ANSI C, finalmente un tutorial che dovrebbe essere molto semplice e dovrebbe scorrere senza troppi problemi. Di che cosa parliamo oggi? Parliamo di variabili, si avete ragione a pensare che ne abbiamo già parlato parecchio in passato, ma in questo caso parliamo non di singole variabili ma di collezioni.

Voi vi starete immaginando le collezioni come quelle delle sorprese degli ovetti della Kinder, oppure come le figurine dei calciatori o i più folli pensano alla loro collezione degli Sgorbions (erano meravigliosi!!!), ebbene non vi state sbagliando, anche qui trattiamo di un insieme di “oggetti” dello stesso tipo, nel nostro caso un insieme di variabili.

Quindi che cosa sono gli array? Sono delle collezioni di variabili dello stesso tipo (int, float, char, etc..) organizzate all'interno di una struttura che le contiene. Immaginiamo l'album delle figurine, l'array non è altro che l'album che le contiene e le figurine sono le singole variabili, guardando le figurine che cosa notiamo di particolare? Esatto sono tutte numerate, ovvero sappiamo esattamente dove si trovano all'interno del nostro contenitore, per gli array vale la stessa cosa: abbiamo una collezione di variabili tutte dello stesso tipo che sono indicizzate in modo tale da trovare sempre la variabile che mi interessa. Ma vediamo come si definisce un array e spieghiamo come funziona:

```
char arrayDiCaratteri[15];
```

Come potete vedere la sua definizione è identica a quella delle variabili, abbiamo il tipo (nel nostro caso char), abbiamo il nome dell'array (in questo caso arrayDiCaratteri, notate che rispetta le convenzioni scelte per i nomi delle singole variabili) ed infine abbiamo un numero tra parentesi quadre. Che cosa indica quel numero? La risposta è abbastanza ovvia, indica il numero di elementi dell'array.

Abbiamo quindi costruito un array di 15 caratteri, nulla di più semplice. Supponiamo di voler scrivere dentro “CORSO DI ANSI C”, avremo una cosa di questo tipo:

|   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |
|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|
| C | O | R | S | O |   | D | I |   | A | N  | S  | I  |    | C  |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

Potete vedere che in ogni cella c'è esattamente un carattere (anche lo spazio è un carattere ASCII) e sotto ad ogni cella vedete l'indice che rappresenta il carattere. Come vedete gli indici vanno da 0 a 14, che sono esattamente 15 elementi come descritto nella definizione dell'array. Bisogna fare molta attenzione e ricordare che la numerazione degli elementi di un array parte da 0, quindi il primo elemento è l'elemento 0.

Supponiamo di voler prendere la lettera A e metterla dentro un'altra variabile, ci basterà fare

```
char nuovaVariabile = arrayDiCaratteri[9];
```

Assegno alla nuova variabile il valore dell'oggetto presente nell'array arrayDiCaratteri all'indice 9, quindi il decimo elemento, se desidero scrivere qualche cosa dentro all'array mi basterà fare una normalissima assegnazione:

```
arrayDiCaratteri[4] = 'A';
```

In questo modo posso riempire il mio array creato con i miei caratteri:

```
char arrayDiCaratteri[15];  
arrayDiCaratteri[0] = 'C';  
arrayDiCaratteri[1] = 'O';  
arrayDiCaratteri[2] = 'R';  
arrayDiCaratteri[3] = 'S';  
arrayDiCaratteri[4] = 'O';
```

Sicuramente scrivere qualche cosa in questo modo non è il massimo della vita, per fortuna abbiamo la possibilità di inizializzare l'array in fase di dichiarazione usando le parentesi graffe:

```
char arrayDiCaratteri[] = { 'C', 'O', 'R', 'S', 'O', ' ',  
                             'D', 'I', ' ', 'A', 'N', 'S',  
                             'I', ' ', 'C' };
```

Ancora non è il massimo vero? Se volete creare un array di numeri siete costretti a mettere le virgole tra un numero e l'altro esattamente come sopra, ma per i caratteri la cosa si può semplificare ulteriormente:

```
char arrayDiCaratteri[] = "CORSO DI ANSI C";
```

Ora ci siamo, decisamente più semplice e veloce. Ora vediamo un piccolo esempio, vogliamo riempire un array di interi con i primi 50 numeri in sequenza, quindi mettere in posizione 0 il numero 0, in posizione 1 il numero 1 e così via. Possiamo fare come abbiamo visto sopra, quindi elencare tutti i numeri a mano separandoli con la virgola oppure possiamo sfruttare il ciclo "for", vediamo come:

```
int arrayDiInteri[50];  
for( int i = 0; i < 50; i++ )  
{  
    arrayDiInteri[i] = i;  
}
```

Ecco fatto, associamo all'elemento in posizione "i" il valore stesso di "i", come abbiamo detto prima il "for" incrementa un contatore, ovviamente questo contatore che noi abbiamo chiamato "i" possiamo usarlo.

Nel prossimo tutorial vedremo il nostro primo vero programma completo, utilizzeremo tutto quello che abbiamo visto in questi 9 tutorial, quindi non perdetelo.

Manuel (aka Jekrom)

per segnalazioni : [manuel.montini@gmail.com](mailto:manuel.montini@gmail.com)

<http://iphonecodes.wordpress.com/>

#### LICENZA:

**Questo tutorial è rilasciato sotto licenza Creative Commons : “Attribuzione-Non commerciale-Non opere derivate 2.5 Italia”,**

<http://creativecommons.org/licenses/by-nc-nd/2.5/it/>