

GLI ENUMERATORI E GLI ALIAS

Il linguaggio C

1



CHE COS'È UN ENUMERATORE

- Un enumeratore permette di definire tipi di dati personalizzati in modo molto flessibile.
- Ad esempio, si possono definire tipi di dati che rappresentano insiemi come:
 - colori;
 - categorie;
 - dimensioni;
 - giorni;
 - mesi;
 - ecc.



COME SI DICHIARA UN ENUMERATORE

La sintassi dell'enumeratore utilizza la parola chiave «**enum**» per definire il nuovo tipo di dato «IdentificatoreTipo» che è rappresentato dall'insieme dei valori inseriti tra le parentesi graffe.

```
enum [<IdentificatoreTipo>] {<Identificatore1>, <IdentificatoreN>} [<Variabile>];
```

```
enum <IdentificatoreTipo> <Variabile>;
```

```
enum colore {bianco, rosso, blu, verde, nero};
```

```
enum colore {bianco, rosso, blu, verde, nero} coloreMuro;
```

```
enum colore coloreMuro;
```



ATTENZIONE...

- Un enumeratore definisce una serie di costanti, raggruppate all'interno di un tipo di dato, che deve essere univoca per l'intero programma.

enum festivo {sabato, domenica};

enum feriale {lunedì, martedì, mercoledì, giovedì, venerdì, sabato};

- Gli identificatori all'interno delle parentesi graffe non sono delle stringhe, per cui non devono essere scritti tra apici doppi.

coloreMuro = bianco;
giorno = sabato;



ATTENZIONE...

- Un enumeratore definisce una serie di costanti numeriche intere che il compilatore, se non diversamente specificato, codifica con valori crescenti a partire da 0.

→ `enum feriale {0lunedì, 1martedì, 2mercoledì, 3giovedì, 4venerdì, 5sabato};`

→ `enum feriale {1lunedì = 1, 2martedì, 3mercoledì, 4giovedì, 5venerdì, 6sabato};`

- E' possibile attribuire a più identificatori valori identici, sia implicitamente che esplicitamente.

→ `enum feriale {0bianco, 1rosso, 0verde = 0, 1blu, 2giallo, 3azzurro, 4rosa};`

- Spesso l'enumeratore è utilizzato per costruire il tipo booleano.

→ `enum bool {falso, vero};`



ESEMPIO

- Calcolo della media dei voti dello studente in italiano, latino, storia e geografia.

```
#include <stdio.h>
#include <stdlib.h>

enum materia {italiano, latino, storia, geografia};

int main(int argc, char *argv[])
{
    float sommaVoti = 0, voto;
    enum materia corso;

    for(corso = italiano; corso <= geografia; corso++)
    {
        printf("Inserisci il voto di ");
```



ESEMPIO

```
switch(corso)
{
  case 0italiano: ←
    printf("italiano: ");
    break;

  case 1latino: ←
    printf("latino: ");
    break;

  case 2storia: ←
    printf("storia: ");
    break;

  case 3geografia: ←
    printf("geografia: ");
    break;
}
```



ESEMPIO

```
scanf("%f", &voto);  
sommaVoti += voto;  
}  
  
printf("\nLa media dei voti e': %.1f sommaVoti / 4);  
  
system("PAUSE");  
return 0;  
}
```



RINOMINARE I TIPI DI DATO

- La parola chiave «**typedef**» (type definition) permette di creare degli alias a partire sia dai tipi fondamentali, sia da quelli strutturati (struct, enum, ecc.).

→ `typedef <tipo> <identificatore>;`

`typedef float temperatura;`

`temperatura tempCitta = 18.6;`

`typedef enum {false, vero} booleano;`

`booleano risposta = vero;`

→ `typedef struct`

`{`

`char marca[15];`

`unsigned cilindrata;`

`} automobile;`

`automobile ferrari = {"Ferrari", 4800};`

