

---

# Introduzione al C

# STRUTTURA DI UN PROGRAMMA C

---

- In prima battuta, la struttura di un programma C è definita nel modo seguente:

```
<programma> ::=  
    {<unità-di-traduzione>}  
    <main>  
    {<unità-di-traduzione>}
```

- *Intuitivamente un programma in C è definito da tre parti:*
  - zero o più unità di traduzione,
  - il programma vero e proprio (main)
  - zero o più unità di traduzione

# STRUTTURA DI UN PROGRAMMA C

---

- La parte `<main>` è l'unica *obbligatoria*, ed è definita come segue:

`<main> ::= main() <blocco>`

`<blocco> ::=`

```
{ [<dichiarazioni-e-definizioni>]
  [<sequenza-istruzioni>]
}
```

- *Intuitivamente il `<main>` è definito dalla parola chiave `main()` e racchiuso tra parentesi graffe al cui interno troviamo*

*•le dichiarazioni e definizioni  
•una sequenza di istruzioni* } *opzionali [ ]*

# STRUTTURA DI UN PROGRAMMA C

---

- **<dichiarazioni-e-definizioni>**

*introducono i nomi* di costanti,  
variabili, tipi definiti dall'utente

- **<sequenza-istruzioni>**

sequenza di frasi del linguaggio  
ognuna delle quali è un'istruzione

**Il `main()` è una particolare unità di traduzione (una funzione).**

# Commenti

---

- I **commenti** servono per inserire un testo all'interno del programma. Sono spiegazioni che vengono date a chi legge il programma. Vengono scartati dal compilatore (non viene prodotto codice eseguibile per i commenti). E' importante mettere commenti nel programma per spiegare che cosa fa (altrimenti anche l'autore dopo un po' non capisce più che cosa fa il programma).
- **Sintassi**: sequenze di caratteri racchiuse fra i delimitatori `/*` e `*/` oppure da `//` fino alla fine della riga  
`<Commento> ::= /* <frase> */`  
`| // <frase> \n`  
`<frase>  ::= { <carattere> }`
- i commenti non possono essere innestati (ma dipende dal compilatore).

# La prima istruzione

---

- Sintassi semplificata:

```
printf("testo");
```

visualizza il testo riportato fra virgolette

- il punto e virgola serve per terminare l'istruzione: in una sequenza di istruzioni, il C capisce dove ne finisce una e comincia la successiva grazie al punto e virgola
- **printf** non è un'istruzione predefinita del C, ma è definita nella libreria **stdio**. Come vedremo, per usare le librerie si utilizza la direttiva **#include**.

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

che va inserita all'inizio del programma

- Si possono usare i caratteri di controllo:

```
newline      \n      tab      \t
backspace    \b
carriage return \r
```

# Il primo programma C

---

```
#include <stdio.h>
/* programma che visualizza la
   scritta "hello world" */
```

```
main ()
```

```
{
```

```
    printf("hello world\n");
```

```
}
```



**Indentazione:** per far capire che `printf` è dentro al `main`, lasciamo un po' di spazio (allineiamo le cose che sono allo stesso livello)