TUTORIAL C LANGUAGE

#2 Episodio: Variabili

- Concetto fondamentale dato che sono uno dei tasselli fondamentali di un codice.

Esempio banale:

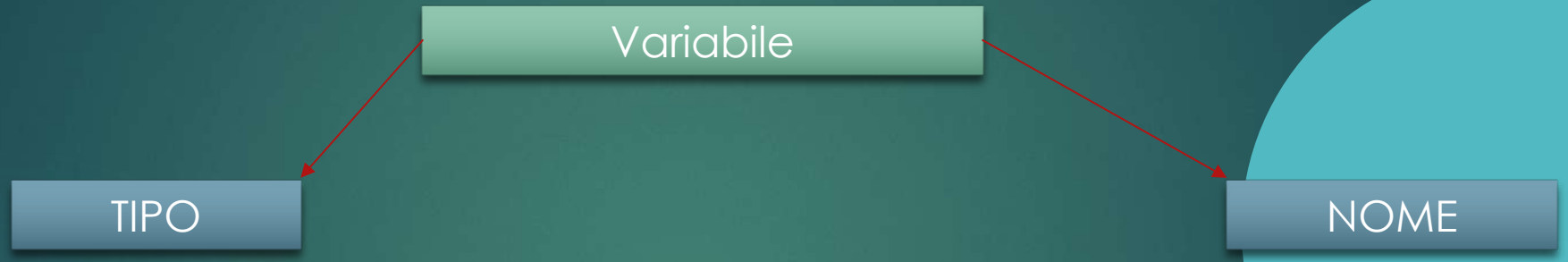
Salvare numero in rubrica.

Nome --> Mario
Numero → 0123 456789

Quindi le variabili sono dei contenitori aventi un nome univoco che può assumere dei valori.

TUTORIAL C LANGUAGE

#2 Episodio: Variabile



Int, char, long, float ecc ….

Int a = 7 e char a = '7', sono diversi infatti basta guardare la tabella ASCII.

Ovviamente è possibile convertire un determinato tipo di variabile in un altro attraverso un'operazione chiamata Casting.

Può essere composto da una o più lettere. Il nome è valido se inizia con una lettera o con un underscore. La loro lunghezza massima dipende dal compilatore, ma di solito non si possono superare i 31 caratteri.
Inoltre il C è Case Sensitive.

Tutte le variabili devono essere dichiarate prima di essere utilizzate, in moda da consentire al compilatore di allocare una determinata memoria per ogni variabile.

TUTORIAL C LANGUAGE

#2 Episodio: Tipi di dati

Esistono diversi tipi di dati ed è molto importante conoscerli attentamente in modo da poter dichiarare quella corretta così da gestire al meglio la memoria a nostra disposizione.

TIPO	RAPPRESENTAZIONE	BYTE	BIT
CHAR	CARATTERE	1	8
INT	NUMERO INTERO	2/4	16/32 (*)
SHORT	NUMERO INTERO CORTO	2	16
LONG	NUMERO INTERO LUNGO	4	32/64
FLOAT	NUMERO REALE	4	32
DOUBLE	NUMERO REALE LUNGO	8	64

* **NOTA**: per alcuni compilatori più obsoleti il tipo int è ancora a 16 bit. Invece nella versione C99 dello standard il linguaggio prevede che int sia a 32 bit. Quest'ultima è la configurazione presente negli odierni compilatori.

TUTORIAL C LANGUAGE

#2 Episodio: Tipi di dati

CHAR

Può contenere qualsiasi lettera e numero da 0 a 9, secondo la tabella ASCII.
Un carattere per volta può contenere UNO ed UN SOLO carattere; in caso si vogliamo memorizzare più carattere in seguito vedremo gli array.

INT

Un numero intero può essere di due tipi:
- Short
- Long

L'int di per se è già uno short, invece il long ci permette di lavorare con numeri più grandi.

```
Int x = 5;
Int y = 3;
Int z;
```

z = x/y = 1 → Ci viene restituita solo la parte intera.

FLOAT & DOUBLE

Numeri in virgola mobile.
Possiamo rappresentare:
- Numeri molto piccoli
- Numeri molto grandi
- Numeri positivi
- Numeri negativi
- Numeri con o senza i decimali

Il differente numero di bit che hanno questi due tipi di dati si riflette sul range di numeri e sul numero di cifre dopo la virgola che possiamo rappresentare.

Quindi se abbiamo bisogno di accuratezza è meglio usare il double.

```
double a = 5.0;
double b = 3.0;
double c;
c = a/b = 1.6
```

NOTA: viene inserito il punto invece della virgola come vuole la notazione inglese.

TUTORIAL C LANGUAGE

#2 Episodio: Tipi di dati

FLOAT & DOUBLE → NUMERI NEGATIVI → COMPLEMENTO A DUE

MSB = 0 → NUMERO POSITIVO
MSB = 1 → NUMERO NEGATIVO

DATO UN NUMERO COME 5, RAPPRESENTATO CON 8 BIT:

0000 0101 → VADO PRIMA AD INVERTIRE I SUOI BIT E POI CI SOMMO 1, RISULTATO:
1111 1011.

TUTORIAL C LANGUAGE

#2 Episodio: Operatori

ARITMETICI

- +
- -
- *
- /
- %

Ovviamente quando si lavora con tipi come il double ed il float l'operatore % non ha senso di essere utilizzato.

Poi abbiamo:
- ++
- --

Questi si possono o anteporre che postporre.

Inoltre:

Espr_1 = Espr_1 <operatore> Espr_2

|
V

Espr_1 <operatore> = Espr_2;

CONFRONTO

Permettono la verifica di determinate condizioni.
- ==
- !=
- <
- >
- <=
- >=

Restituiscono TRUE se è verificata la condizione, FALSE se non lo è.

Questi operatorio saranno utili quando lavoreremo con gli operatori condizionali.

LOGICI

Anche questi utilizzati con le istruzioni condizionali ed iterative e permettono di eseguire la AND e OR tra operandi:

- && → = 1 se A = B = 1
- || → = 1 se A o B = 1
- NOT

Ovviamente son diversi da & e | che sono operatori bitwise che servono proprio per eseguire l'operazione logica.

TUTORIAL C LANGUAGE

#2 Episodio: Proprietà Operatori

POSIZIONE

- PREFISSO: se compare prima dell'operando
- POSTFISSO: se compare dopo gli operandi
- INFISSO: se compare tra gli operandi

ARIETA'

Numero di argomenti che un operatore può accettare:

- ++: Arietà = 1
- +: Arietà = 2

PRECEDENZA/PRIORITA'

Identifica gli operandi che hanno il diritto di essere eseguiti per prima.

ASSOCIATIVA

A parità di priorità stabilire quale sia l'ordine con cui bisogna eseguire i vari operatori:

- SX → DX
- DX → SX