

Semplice

Input/output in JAVA

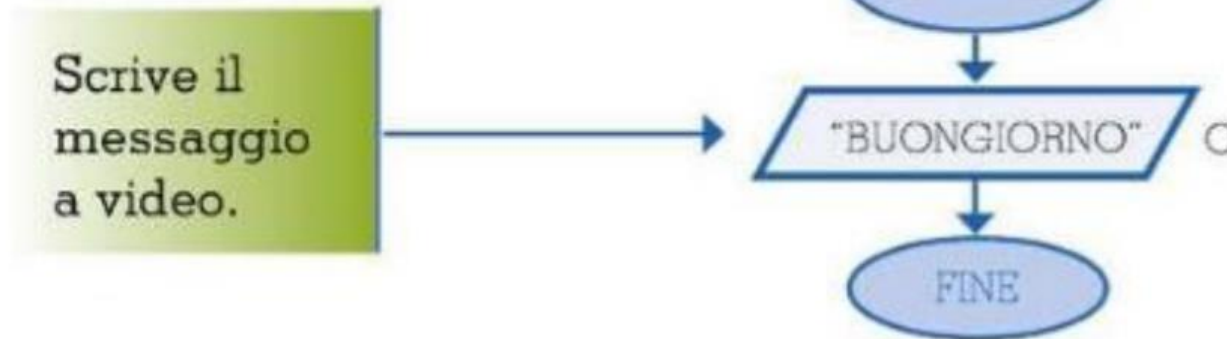
Primi passi con JAVA

Stampa a video il saluto Buongiorno

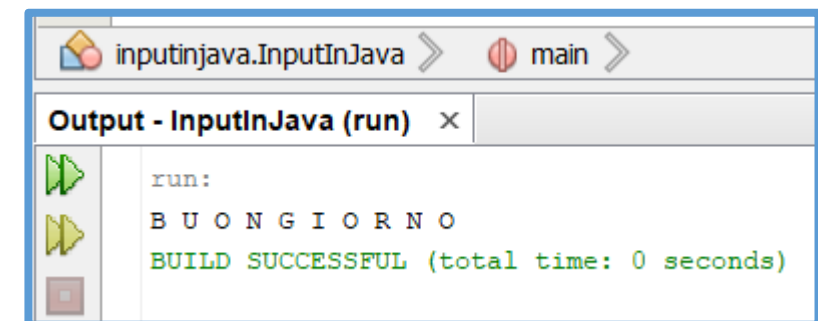


Testo Stampa a video il saluto *Buongiorno*.

ALGORITMO



```
public class StampaBuongiorno{  
  
    public static void main(String[] args) {  
  
        System.out.println("B U O N G I O R N O");  
    } //main  
} //class
```



La classe Scanner

La classe **Scanner** rappresenta la soluzione più semplice per realizzare in Java un input da uno stream di caratteri.

La classe **Scanner** appartiene al package *java.util* e i suoi oggetti sono in grado di individuare e interpretare *i tipi primitivi e le stringhe* presenti all'interno di uno stream di caratteri.

La classe Scanner suddivide lo stream dei caratteri in **token**, cioè in *spezzoni di stringhe* separate dai *caratteri delimitatori*. I caratteri delimitatori di default sono: *gli spazi, i caratteri di tabulazione e i caratteri di newline*).

int nextInt(): legge il token successivo interpretandolo come un numero intero e lo restituisce al chiamante, o lancia un'eccezione di tipo `InputMismatchException` se il token non è un intero.

double nextDouble(): legge il token successivo interpretandolo come un numero reale e lo restituisce al chiamante, o lancia un'eccezione di tipo `InputMismatchException` se il token non è un reale.

String nextLine(): legge la riga di testo (successiva) e la restituisce al chiamante.

String next(): legge il successivo token senza delimitatori e lo restituisce al chiamante.

.....

Input/output di variabili alfanumeriche

Testo Stampa un messaggio di saluto per una persona il cui nome è fornito in input.

Analisi Per risolvere questo problema devi richiedere in *input* il nome di una persona e restituirlo in *output* insieme a un messaggio predefinito.



```
import java.util.Scanner;

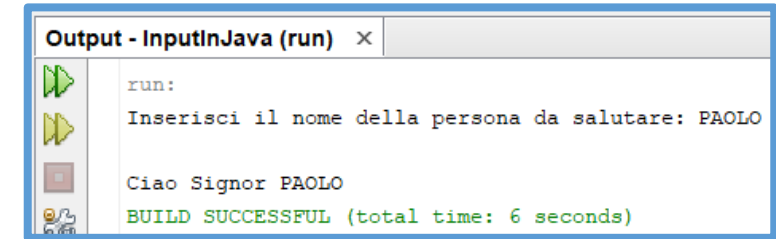
public class InputInJava {

    public static void main(String[] args) {
        String nome;
        Scanner tastiera = new Scanner(System.in);

        System.out.print("Inserisci il nome della persona da salutare: ");
        nome = tastiera.nextLine();

        System.out.println("\nCiao Signor "+nome);

    } //main
} //class
```



Somma di due numeri interi

```
import java.util.Scanner;

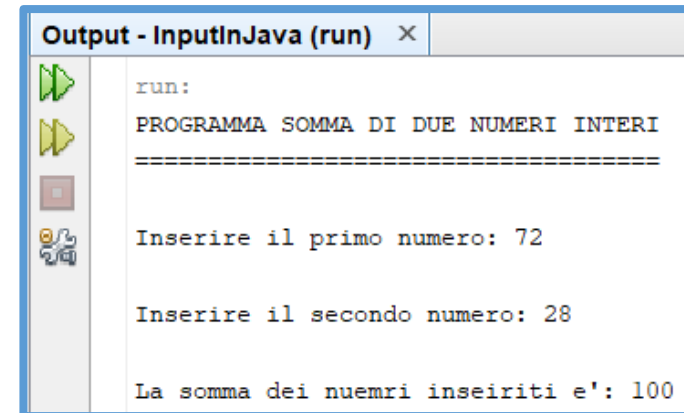
public class InputInJava {
    public static void main(String[] args) {
        int a, b, somma;
        Scanner tastiera = new Scanner( source: System.in);
        System.out.println( x: "PROGRAMMA SOMMA DI DUE NUMERI INTERI");
        System.out.println( x: "=====");
        System.out.println();

        System.out.print( s: "Inserire il primo numero: ");
        a = tastiera.nextInt();

        System.out.print( s: "\nInserire il secondo numero: ");
        b = tastiera.nextInt();

        somma = a+b;

        System.out.print("\nLa somma dei nuemri inseiriti e': "+somma);
        System.out.println();
    } //main
} //class
```



```
Output - InputInJava (run) x
run:
PROGRAMMA SOMMA DI DUE NUMERI INTERI
=====
Inserire il primo numero: 72
Inserire il secondo numero: 28
La somma dei nuemri inseiriti e': 100
```

I tipi di dato in JAVA

TIPI DI DATI (primitivi) IN JAVA si suddividono in

	NOME IDENTIFICATIVO	Tipo di dato	Occupazione in memoria
Booleani	boolean	Valori logici (true/ false)	Non specificato ma basterebbe 1 bit
Caratteri	char	Carattere	2 Bytes
Interi	byte	Valori interi	8 Bytes
	short	Valori interi	16 Bytes
	int	Valori interi	32 Bytes
	long	Valori interi	64 Bytes
Reali	float	Valori Reali (in virgola mobile)	32 Bytes
	double	Valori Reali (in virgola mobile)	64 Bytes

```
int x = 5;
int y = 10;
boolean risultatoBinario;

risultatoBinario = (x == y);
System.out.println(risultatoBinario); // restituisce false

risultatoBinario = (x < y);
System.out.println(risultatoBinario); // restituisce true
```

Le stringhe

Una particolare attenzione dobbiamo dare all'oggetto **String** che contiene una serie di char, ma non essendo un Tipo di dato SEMPLICE andrà dichiarato utilizzando la parola chiave **new**. Dunque dato che String deve essere istanziato come un oggetto, si capisce che il Java ci mette a disposizione la **classe String**. Vediamo l'utilizzo dell'oggetto string:

```
// esempio di oggetto String
String OggettoStringa = new String("Valore testuale ");

OggettoStringa = OggettoStringa + "dell'OggettoStringa";

System.out.println(OggettoStringa);
```

Avremo come risultato su schermo "**Valore testuale dell'OggettoStringa**" perchè abbiamo utilizzato l'operatore **+** per concatenare le due stringhe.

Operatori Aritmetici

Nelle istruzioni di assegnazione possono comparire gli operatori aritmetici per eseguire dei calcoli.

I cinque operatori aritmetici previsti dal linguaggio sono:

operatore	significato
+	somma
-	differenza
*	moltiplicazione
/	divisione
%	modulo (o resto)

Le conversioni (casting)

Ciascun operatore **DEVE** OPERARE CON TIPI DI DATO OMOGENEI. Quando vogliamo lavorare con variabili di tipo simili, per esempio per sommare una variabile di tipo **int** ad una di tipo **short**, avviene automaticamente una conversione: il valore **short** viene promosso a **int** in modo da eseguire l'operazione su dati omogenei. Questo concetto è chiamato **PROMOZIONE NUMERICA** e viene eseguito da JAVA in maniera **automatico** tra tipi di omogenei, quindi interi con interi oppure decimali con decimali

```
// esempio di promozione automatica
int a = 5;
long l = a;
```

Se succede qualcosa del tipo:

```
float numeroFloat = 2.5;
int i = numeroFloat ;
```

In questo caso il compilatore ci avverte che può esserci una perdita di precisione (informazione) visto che il tipo di dato **float** può contenere valori più grandi di quelli memorizzabili in un **int**. JAVA chiede esplicitamente al programmatore di specificare esplicitamente che l'operazione scritta nel programma sia effettivamente quella che intendeva realizzare (e non per esempio un banale errore di programmazione). L'operazione che il programmatore è chiamato a fare si chiama **casting**

Il casting è possibile tra tipi di dati compatibili, non è possibile eseguire il casting tra una stringa ed un numero, un carattere ed un booleano. In generale i numeri reali e interi, di qualsiasi dimensioni sono convertibili tra di loro, ma non con altri tipi.

```
// esempio Casting
float f = 2.5;
\
int i = (int) f;
// converte f in un intero escludendo i
// decimali dichiarato con i
```

Operatori di confronto *(piccolo anticipo dell'if)*

= assegnazione
== ugualianza
!= differenza
> maggiore
< minore
&& e (AND logico)
|| o (OR logico)

>= maggiore e uguale
<= minore e uguale
+= assegna e aggiunge un valore
-= assegna e sottrae un valore

```
int x = 1;           // utilizziamo l'operatore = di assegnazione
int y = x + 1;       // y sarà uguale a 2
y += 10;             // y diventa 12
```

```
if (x == y) {
    System.out.println("Primo risultato FALSO");
}else if (y > x){
    System.out.println("Secondo risultato VERO");
}
```