

*"I think everybody should learn how to program a computer, because it teaches you how to think."*

*"Tutti dovrebbero imparare a programmare un computer perché è un'attività che ti insegna a pensare."*

*(Steve Jobs, fondatore di Apple 🍏)*

# L'informatica e la programmazione

- Corso per la Scuola Media  
e Media Superiore -



# Indice dei contenuti

<b>1</b>	<b>INTRODUZIONE: COSA È L'INFORMATICA</b>	<b>1</b>
<b>2</b>	<b>LA PROGRAMMAZIONE DI UN COMPUTER</b>	<b>3</b>
<b>3</b>	<b>LABORATORIO 1: PROGRAMMAZIONE STRUTTURATA CON LOGO</b>	<b>4</b>
<b>3.1</b>	<b>SEQUENZE DI ISTRUZIONI, PROCEDURE E PARAMETRI</b>	<b>6</b>
<b>3.2</b>	<b>CICLI</b>	<b>8</b>
<b>3.3</b>	<b>SELEZIONI O ISTRUZIONI CONDIZIONALI</b>	<b>9</b>
<b>3.4</b>	<b>CONCLUSIONE DEL LABORATORIO 1</b>	<b>11</b>
<b>3.5</b>	<b>APPROFONDIMENTO SU CICLI E PROCEDURE RICORSIVE (FACOLTATIVO)</b>	<b>12</b>
<b>3.6</b>	<b>APPROFONDIMENTO SU ISTRUZIONI CONDIZIONALI E LOGICA BOOLEANA (FACOLTATIVO)</b>	<b>16</b>
<b>4</b>	<b>LABORATORIO 2: NIBBLES/SNAKE, IL MIO PRIMO VIDEOGIOCO</b>	<b>18</b>
<b>4.1</b>	<b>SVILUPPARE UN'APPLICAZIONE</b>	<b>18</b>
<b>4.2</b>	<b>ANALISI DEI REQUISITI</b>	<b>18</b>
<b>4.3</b>	<b>CONCEZIONE</b>	<b>20</b>
4.3.1	FUNZIONAMENTO GENERALE (E ANIMAZIONI GRAFICHE)	20
4.3.2	FUNZIONAMENTO DETTAGLIATO	21
<b>4.4</b>	<b>IMPLEMENTAZIONE E TEST</b>	<b>23</b>
4.4.1	GRAFICA	23
4.4.2	VARIABILI ED ELENCHI	25
4.4.3	PROCEDURE: VALORI DI RITORNO	27
4.4.4	INPUT UTENTE	29
4.4.5	ANIMAZIONE	30
4.4.6	CONCLUSIONE DELL'ATTIVITÀ DI IMPLEMENTAZIONE	30
<b>4.5</b>	<b>COLLAUDO</b>	<b>30</b>
<b>4.6</b>	<b>CONCLUSIONE DEL LABORATORIO 2</b>	<b>31</b>
	<b>RIFERIMENTI</b>	<b>33</b>
	<b>APPENDICE: VOCABOLARIO DELLE ISTRUZIONI DI LOGO</b>	<b>34</b>

# 1 Introduzione: cosa è l'informatica

La parola "informatica" proviene dal francese "informatique" e, più precisamente, dalla contrazione della definizione "**in**formation **automatique**", che indica l'elaborazione automatica dell'informazione.

Molte persone confondono l'informatica con il semplice utilizzo del computer o di applicazioni quali editor di testo (es. *Office Word*), browser per la navigazione in rete (*Firefox*, *Internet Explorer*, *Safari*, ...), social network (*Facebook*, *Twitter*, *Google+*,...) e videogiochi (...).

In realtà, l'informatica studia come automatizzare in modo efficiente (ovvero *ottimizzare*) un **processo** che trasformi una determinata richiesta o dato iniziale (**input**) in un risultato desiderato (**output**), avendo a disposizione un certo insieme di **risorse**.



"...uff... ma quante parole strane... cerchiamo di capire con un esempio..."

Immaginiamo di voler spiegare ad un compagno come si disegna un quadrato con lato pari a 5cm, avendo a disposizione **risorse** quali *riga*, *matita* e *foglio* di carta bianco. Cosa dobbiamo fare? Scrivi qui sotto un elenco ordinato delle operazioni da eseguire.

Metodo (ossia ...**processo**) per disegnare un quadrato con lato pari a 5cm:

Dunque... cosa abbiamo fatto? Spiega...



"...uhmm... ma così il quadrato non è molto preciso...". Questo processo non è efficiente!

Come possiamo migliorare questo processo, magari aggiungendo una nuova **risorsa** quale il compasso?

Metodo ottimizzato per disegnare un quadrato di lato pari a 5cm con il compasso:  
(Se non sai come si fa, prova a cercare in internet inserendo le parole chiave "quadrato con compasso")

Dunque... cosa abbiamo fatto? Spiega...

---

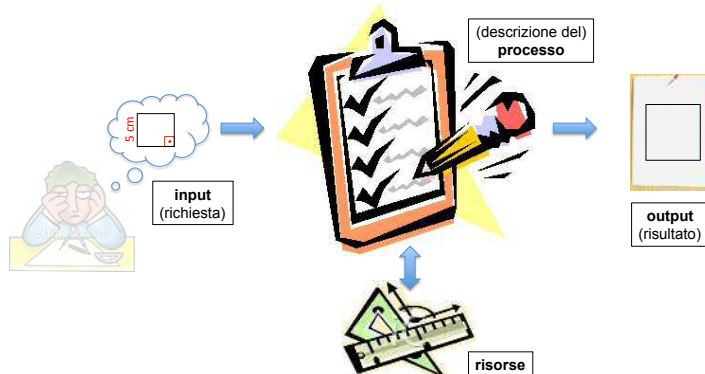
---

---



"wow... adesso sì che il quadrato è preciso!"

Dunque questo significa **automatizzare** e **ottimizzare** un **processo**, ovvero ciò che fanno gli informatici!



Riassumendo, definiamo con parole nostre i concetti che abbiamo visto e diamo alcuni esempi:

*Processo:*

---

---

---

---

*Input:*

---

---

---

---

*Output:*

---

---

---

---

*Risorsa:*

---

---

---

---

## 2 La programmazione di un computer

Programmare un computer significa insegnare un *processo* a una macchina, che altrimenti sarebbe completamente inutile. In pratica, dobbiamo fargli capire cosa fare attraverso delle istruzioni precise, esattamente come abbiamo fatto in precedenza per costruire il quadrato.

Il computer però non è in grado di capire la nostra lingua e dunque non possiamo spiegargli cosa deve fare come se parlassimo a un nostro compagno. Dobbiamo invece usare un **linguaggio** in cui esiste un numero limitato di parole (o, meglio, **istruzioni**). Inoltre, questo linguaggio deve osservare determinate regole, una **grammatica**, esattamente come succede in ogni lingua. Imparare un linguaggio di programmazione è dunque molto simile a imparare una lingua straniera.

Il modo classico di impartire le istruzioni necessarie a completare un processo è la scrittura di un **codice** nel quale sono elencate tutte le operazioni necessarie. Oggigiorno i programmi si possono scrivere direttamente al computer con dei semplici editor di testo.



### 3 Laboratorio 1: programmazione strutturata con Logo

Nel nostro Laboratorio impareremo come scrivere un programma per mezzo del linguaggio **Logo**. Come per ogni altra lingua, è importante impararne più parole (istruzioni) possibili e soprattutto la grammatica. Basta infatti un piccolo errore e il computer non è più in grado di capire! Un buon programmatore deve essere in grado di "parlare" con il computer in modo chiaro e preciso, facendogli eseguire i compiti in modo efficiente.



#### Compito 1: Download e avvio dell'applicazione XLogo

Per poter eseguire delle istruzioni nel linguaggio *Logo* è necessario scaricare e avviare l'applicazione *XLogo*.

- Avvia un browser (*Internet Explorer, Firefox, ...*) e inserisci l'indirizzo:  
<http://downloads.tuxfamily.org/xlogo/common/xlogo.jar>  
 (in alternativa è possibile inserire la parola chiave "xlogo" in un motore di ricerca quale *Google, Bing, ...* e selezionare il risultato opportuno).

Si aprirà una "finestra di dialogo" che ti chiederà se vuoi scaricare il file *xlogo.jar*: accetta e salva il file sul *Desktop*.

- Chiudi il browser, clicca due volte velocemente sull'icona del file scaricato e **aspetta le prossime istruzioni**.

Il linguaggio *Logo* "standard" utilizza delle parole inglesi per definire le istruzioni che il computer può capire. Ad esempio per far scrivere 'Buongiorno' al computer dovremmo digitare la seguente istruzione:

Write "Buongiorno"

L'applicazione *XLogo* offre però la possibilità di "tradurre" tutte queste parole in italiano. Per questo, la finestra di dialogo iniziale ci chiederà di indicare la nostra lingua. Selezionate dunque sulla bandiera italiana e cliccate su *OK* (il bottone *OK* è visibile facendo scorrere il contenuto della finestra verso il basso).

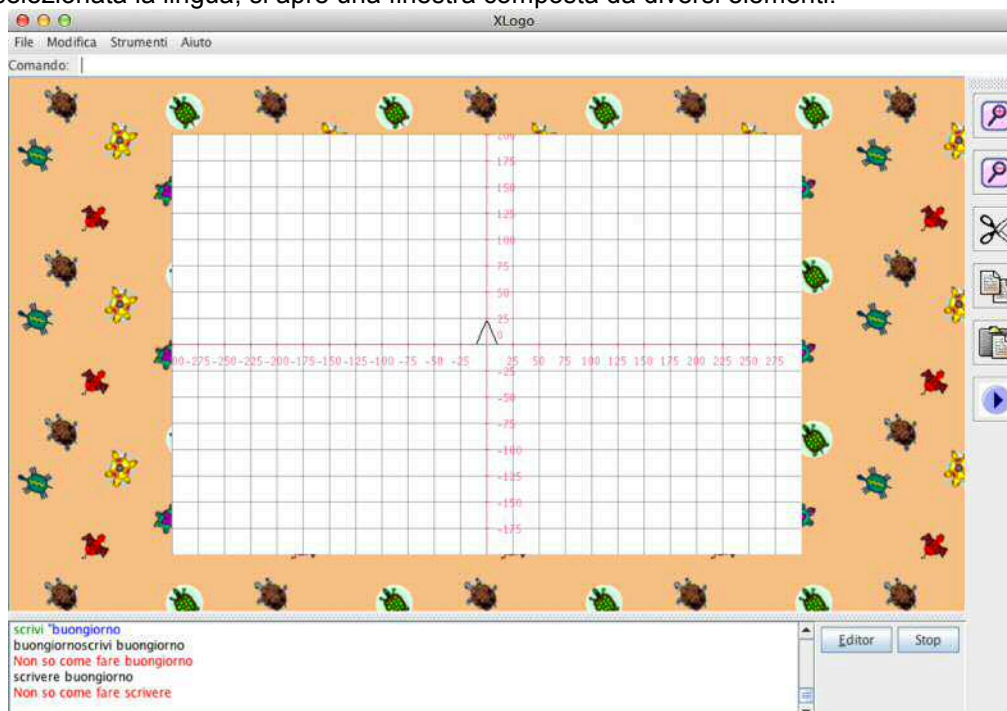


Adesso ci basterà scrivere l'istruzione:

**Scrivi "Buongiorno"**

e il computer sarà in grado di capire le parole in italiano che corrispondono a delle istruzioni.

Una volta selezionata la lingua, si apre una finestra composta da diversi elementi:



Nella parte superiore, sotto la barra del menu (*File, Modifica, Strumenti, Aiuto*), è presente una linea di *Comando* nella quale si scrivono le istruzioni per il computer.

Nella parte centrale, abbiamo invece piano cartesiano in cui è possibile disegnare dando al computer le istruzioni opportune. L'origine del piano (0,0) è situata al centro, mentre la posizione della "penna/tartaruga" è rappresentata dal triangolo.

Nella parte inferiore sono elencate tutte le istruzioni date in precedenza al computer (si chiama "*lo storico dei comandi*") come pure le risposte. Per esempio se dovessimo impartire il comando:

*Scrivi "Buongiorno*

Nello storico vedremo:

*Scrivi "Buongiorno*  
Buongiorno

Le risposte del computer sono scritte in **nero**, mentre i messaggi di errore sono scritti in **rosso**. Le istruzioni (primitive) sono indicate in **verde** e i valori (numeri, parole, variabili, ...) sono colorati di **blu**.



*Compito 2: La nostra prima istruzione Logo: facciamo scrivere Buongiorno al computer.*

- *Inserisci l'istruzione opportuna nella riga di comando.*
- *Cosa succede se dovessimo inserire l'istruzione:  
Scrivere "Buongiorno  
Perché? Spiega cosa risponde il computer...*

- *Cosa succede se dovessimo inserire l'istruzione:  
Scrivi Buongiorno  
Perché? Spiega cosa risponde il computer...*

*Dunque... cosa abbiamo visto?*

### 3.1 Sequenze di istruzioni, procedure e parametri

Come abbiamo visto nel *Capitolo 1*, i processi sono descritti attraverso delle (**sequenze di**) istruzioni.



*Compito 3: Il nostro primo processo in Logo: facciamo disegnare un quadrato al computer con il lato di 90 passi.*

Per questo compito dobbiamo conoscere due nuove istruzioni:

**Avanti** `passi`

Muove la penna in avanti di un numero di passi pari al valore indicato con `passi`<sup>1</sup>. La direzione in cui si muove la penna è indicata dalla punta più acuta del triangolo.

**RuotaDestra** `gradi`

Fa ruotare di `gradi` gradi la direzione della penna.

- *Scrivi qui sotto il codice completo (ovvero la sequenza di istruzioni che vi permettono di disegnare il quadrato richiesto). Se provando ad inserire i comandi fai un errore, puoi cancellare lo schermo con l'istruzione **PulisciSchermo**.*

*Dunque... cosa abbiamo visto?*

Ogni volta che vogliamo disegnare un quadrato, dovremmo però ripetere tutta questa sequenza di istruzioni. Fortunatamente i linguaggi di programmazione permettono di definire delle **procedure**. Le procedure sono delle sequenze di istruzioni che possono essere richiamate ogni volta che è necessario, come se fossero delle istruzioni primitive proprie del linguaggio di programmazione.

<sup>1</sup> **Attenzione:** Quando, spiegando le istruzioni, usiamo una parola in evidenza, dobbiamo rimpiazzare quanto da essa descritto con dei valori opportuni. Per esempio, `passi` significa che dovrete scrivere il numero di passi voluti: 10, 20, ... Un'istruzione valida potrebbe dunque essere: **Avanti** `20`





**Compito 4:** Scriviamo una procedura chiamata 'Quadrato' per disegnare un quadrato con lato di 90 passi.

Per questo compito dobbiamo conoscere due nuove istruzioni:

**Per** `nomeProcedura`

Apri una finestra di editor nel quale si definisce la procedura `nomeProcedura`. Ogni procedura ha la seguente struttura:

```

Per nomeProcedura
sequenzaIstruzioni
Fine

```

Per abilitare la procedura bisogna cliccare sul bottone in alto a sinistra (il triangolo).

**Modifica** [`nomeProcedura`]

Per modificare la procedura già definita (attenzione alle parentesi quadre!).

- *Scrivi il codice della procedura qui sotto. Come facciamo a lanciarla?*

*Dunque... cosa abbiamo visto?*

Ora che abbiamo definito la procedura 'Quadrato', ci piacerebbe poter disegnare dei quadrati con lati di misura diversa. Questo costituisce però un problema, perché, come di programmatori, dovremmo scrivere un numero infinito di procedure (processi) che disegnino quadrati per tutte dimensioni possibili. Fortunatamente però, le procedure possono essere **parametrizzate**: anziché definire a-priori i valori che saranno utilizzati, possiamo rappresentarli in modo generale attraverso dei parametri, come avviene ad esempio nelle *formule algebriche*. Dunque, nel nostro esempio, non utilizzeremo più l'istruzione `Avanti 90`, bensì `Avanti :a`, dove `a` è il parametro indicante la misura del lato e `:` un operatore. In Logo, le procedure parametrizzate si definiscono aggiungendo il nome del parametro, preceduto dall'operatore, accanto al nome della procedura (es. `Per Quadrato :a`) e vengono lanciate indicando il valore che questo parametro dovrà assumere (es. `Quadrato 90`). Allo stesso modo è possibile definire più parametri per ogni procedura.



*Compito 5: Scriviamo una procedura parametrizzata per disegnare un quadrato con dimensioni variabili (modificate la procedura scritta precedentemente).*

- *Scrivi il codice della procedura qui sotto. Come facciamo a lanciarla?*

*Dunque... cosa abbiamo visto?*

Il nome assegnato ai parametri non deve necessariamente essere composto da una sola lettera. Allo scopo di rendere il codice più comprensibile, i programmatori utilizzano spesso dei nomi più illustrativi. Ad esempio anziché definire il nostro parametro come `a` (`Quadrato :a`) potremmo chiamarlo `:misuraLato`. La nostra procedura sarebbe quindi definita come:

`Per Quadrato :misuraLato`

ciò che renderebbe chiaro l'utilizzo di questo parametro a chi legge il codice. **D'ora in poi useremo sempre dei nomi appropriati per i parametri!**

### 3.2 Cicli

A questo punto abbiamo imparato a scrivere una procedura parametrizzata. Però, guardando il codice, ci accorgiamo che ci sono un sacco di ripetizioni ...e, fortunatamente, si tratta solo di un quadrato. Se fosse un poligono con più lati, ad esempio 50 lati, dovremmo ripetere 50 volte le stesse istruzioni. Per fortuna i linguaggi di programmazione consentono di utilizzare una scorciatoia e far fare al computer le ripetizioni necessarie, anziché riscrivere tante volte le stesse istruzioni. Le sequenze di istruzioni ripetute vengono comunemente dette **cicli**.

# Il corso completo è disponibile gratuitamente previo contatto

<http://insegnamento.adamoli.ch>