Fondamenti di programmazione

I ***fondamenti di programmazione*** sono quelle conoscenze basilari che occorre possedere per la [programmazione (informatica)](https://it.wikiversity.org/w/index.php?title=Programmazione_(informatica)&action=edit&redlink=1) in un qualunque [linguaggio di programmazione](https://it.wikiversity.org/w/index.php?title=Linguaggio_di_programmazione&action=edit&redlink=1).

Per scrivere le righe di codice dei [linguaggi di programmazione](https://it.wikiversity.org/w/index.php?title=Linguaggi_di_programmazione&action=edit&redlink=1) è sufficiente utilizzare un normale [elaboratore testi](https://it.wikiversity.org/w/index.php?title=Elaboratore_testi&action=edit&redlink=1), o un programma ad hoc detto [editor](https://it.wikiversity.org/w/index.php?title=Editor&action=edit&redlink=1). Nel secondo caso il [software](https://it.wikiversity.org/w/index.php?title=Software&action=edit&redlink=1) individua gli errori più banali che un [programmatore](https://it.wikiversity.org/w/index.php?title=Programmatore&action=edit&redlink=1) può commettere.

# Le Fasi della programmazione

I passi con cui si sviluppa un programma efficiente sono simili tra di loro per quasi tutti i contesti di [programmazione (informatica)](https://it.wikiversity.org/w/index.php?title=Programmazione_(informatica)&action=edit&redlink=1) (aziende, hobby, lavoro, ecc.). Le prime fasi sono strettamente legate alla [logica](https://it.wikiversity.org/wiki/Logica) di quello che si va a creare e del suo funzionamento, mentre le ultime sono orientate verso lo specifico [linguaggio di programmazione](https://it.wikiversity.org/w/index.php?title=Linguaggio_di_programmazione&action=edit&redlink=1) che si intende utilizzare.

## Il problema da risolvere

La prima operazione da svolgere per la creazione di un qualunque [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) è definire nel modo più preciso ed accurato possibile quale problema intendiamo risolvere. Questo vale sia per una semplice [funzione matematica](https://it.wikiversity.org/w/index.php?title=Funzione_matematica&action=edit&redlink=1) che per un complesso [database](https://it.wikiversity.org/w/index.php?title=Database&action=edit&redlink=1). L'[input](https://it.wikiversity.org/w/index.php?title=Input&action=edit&redlink=1) che il [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) riceve è legato all'[output](https://it.wikiversity.org/w/index.php?title=Output&action=edit&redlink=1) che intendiamo restituire dopo l'elaborazione.

Questa fase può presentare una grande complessità dovuta al lavoro astratto che impone al [programmatore](https://it.wikiversity.org/w/index.php?title=Programmatore&action=edit&redlink=1) di immaginare il risultato finale e la sua formalizzazione.

## La progettazione del programma

Superata la definizione del problema che si vuole risolvere occorre progettare delle possibili soluzioni, porle a confronto tra loro e risalire a quella più efficiente (ricordando che una soluzione non efficiente per un determinato problema potrebbe essere utile allo sviluppo di altri programmi). Per confrontare l'effettiva differenza tra una soluzione ed un'altra possono essere utilizzati due diversi strumenti:

* Uno di essi è il [diagramma di flusso](https://it.wikiversity.org/w/index.php?title=Diagramma_di_flusso&action=edit&redlink=1): con un [grafico](https://it.wikiversity.org/w/index.php?title=Grafico&action=edit&redlink=1) molto intuitivo si può illustrare un processo in tutte le fasi che lo caratterizzano. Ogni fase è collegata con le altre dalle [linee di flusso](https://it.wikiversity.org/w/index.php?title=Linee_di_flusso&action=edit&redlink=1) che indicano la direzione del processo interno al [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1).
* Un efficace metodo di progettazione è lo [pseudocodice](https://it.wikiversity.org/w/index.php?title=Pseudocodice&action=edit&redlink=1), una forma di scrittura in codice affine alla lingua parlata dal [programmatore](https://it.wikiversity.org/w/index.php?title=Programmatore&action=edit&redlink=1). La sua stesura è simile ad una scaletta di operazioni che si definiscono quando si deve eseguire una determinata azione. La scrittura dello [pseudocodice](https://it.wikiversity.org/w/index.php?title=Pseudocodice&action=edit&redlink=1) deve essere dettagliata almeno quanto il codice finale, ma rispetto ad esso non si richiede una sintassi precisa lasciando il [programmatore](https://it.wikiversity.org/w/index.php?title=Programmatore&action=edit&redlink=1) libero di concentrarsi sui problemi principali del [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1).

## Gli Algoritmi

Al fine di creare un [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) efficiente occorre progettarne con attenzione l'[algoritmo](https://it.wikiversity.org/w/index.php?title=Algoritmo&action=edit&redlink=1). Ad esempio potremmo creare un [algoritmo](https://it.wikiversity.org/w/index.php?title=Algoritmo&action=edit&redlink=1) che esegue i seguenti passi:

1. Esamina una lista di numeri e riconosci quello più basso;
2. Inserisci il numero più basso nella [variabile](https://it.wikiversity.org/w/index.php?title=Variabile&action=edit&redlink=1) X;
3. Stampa la [variabile](https://it.wikiversity.org/w/index.php?title=Variabile&action=edit&redlink=1) X;

È importante precisare che la lista di passi deve contenere istruzioni chiare, precise e non ambigue, ovvero il cui significato non dipenda dal contesto in cui lo si scrive, ma sia chiaramente identificabile.

I programmi possono inoltre essere divisi in sezioni in modo che i vari compiti possano essere eseguiti attraverso sequenze più semplici di passi. A tale scopo viene suddiviso il [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) in [funzioni (informatica)](https://it.wikiversity.org/w/index.php?title=Funzioni_(informatica)&action=edit&redlink=1).

## La codifica del programma

Avviene in questa fase la scelta del [linguaggio di programmazione](https://it.wikiversity.org/w/index.php?title=Linguaggio_di_programmazione&action=edit&redlink=1) con cui sviluppare il progetto ([Java](https://it.wikiversity.org/w/index.php?title=Java_(linguaggio)&action=edit&redlink=1), [C++](https://it.wikiversity.org/wiki/C%2B%2B), [BASIC](https://it.wikiversity.org/w/index.php?title=BASIC&action=edit&redlink=1), [Perl](https://it.wikiversity.org/w/index.php?title=Perl&action=edit&redlink=1), Python ecc.). Occorre passare dai [diagrammi di flusso](https://it.wikiversity.org/w/index.php?title=Diagrammi_di_flusso&action=edit&redlink=1) o dallo [pseudocodice](https://it.wikiversity.org/w/index.php?title=Pseudocodice&action=edit&redlink=1), ad una sintassi più rigorosa e specifica del linguaggio scelto.

L'operazione di [codifica](https://it.wikiversity.org/w/index.php?title=Codifica&action=edit&redlink=1) è quella più soggetta ad errori di natura logica e sintattica. In teoria il [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) potrebbe essere pronto in questa fase, ma in realtà esso presenta nella quasi totalità dei casi degli errori, che verranno analizzati con la fase successiva.

## La verifica del risultato

Completato il [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) è frequente la presenza di errori derivanti da ragioni varie.

La fase di verifica e di messa a punto del [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) è detta [debugging](https://it.wikiversity.org/w/index.php?title=Debugging&action=edit&redlink=1) ed ha lo scopo di ricercare e correggere gli errori riscontrati, detti [bug](https://it.wikiversity.org/w/index.php?title=Bug&action=edit&redlink=1). Si tratta di una fase molto lunga ed importante in cui il [programmatore](https://it.wikiversity.org/w/index.php?title=Programmatore&action=edit&redlink=1) verifica la bontà del proprio lavoro. La correzione di un [bug](https://it.wikiversity.org/w/index.php?title=Bug&action=edit&redlink=1) potrebbe essere causa di nuovi errori, occorre quindi prestarci la massima attenzione.

Per evitare un [debugging](https://it.wikiversity.org/w/index.php?title=Debugging&action=edit&redlink=1) di un [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) troppo complesso ed articolato è preferibile creare un [prototipo](https://it.wikiversity.org/w/index.php?title=Prototipo&action=edit&redlink=1), ovvero una versione del risultato finale che non presenta in sé tutte le funzioni previste dal progetto originale, ottenendo una maggiore velocità di correzione dei [bug](https://it.wikiversity.org/w/index.php?title=Bug&action=edit&redlink=1) principali.

Alla fine di questa fase avremo il prodotto finito. Le operazioni svolte durante lo sviluppo vanno documentate così da rendere più semplice una futura modifica.

# Esempio: addizione di due numeri

Il [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) più semplice ed intuitivo è sicuramente quello che presi due numeri in [input](https://it.wikiversity.org/w/index.php?title=Input&action=edit&redlink=1), restituisce come [output](https://it.wikiversity.org/w/index.php?title=Output&action=edit&redlink=1) la loro somma. A seconda del [linguaggio di programmazione](https://it.wikiversity.org/w/index.php?title=Linguaggio_di_programmazione&action=edit&redlink=1) che si utilizza la struttura cambia e si articola in maniera diversa.

* In un [linguaggio macchina](https://it.wikiversity.org/w/index.php?title=Linguaggio_macchina&action=edit&redlink=1) un determinato comando x1 carica un numero che si trova in una [locazione di memoria](https://it.wikiversity.org/w/index.php?title=Locazione_di_memoria&action=edit&redlink=1) y1 (generalmente definita da una cifra [esadecimale](https://it.wikiversity.org/w/index.php?title=Esadecimale&action=edit&redlink=1)) all'interno della [CPU](https://it.wikiversity.org/wiki/CPU). Il comando x2 somma il numero che si trova nella [locazione di memoria](https://it.wikiversity.org/w/index.php?title=Locazione_di_memoria&action=edit&redlink=1) y2 con quello caricato precedentemente. Infine un comando x3 trasferisce il risultato ottenuto in una [locazione di memoria](https://it.wikiversity.org/w/index.php?title=Locazione_di_memoria&action=edit&redlink=1) y3 rendendolo disponibile per nuove operazioni.
* In un [linguaggio assembly](https://it.wikiversity.org/w/index.php?title=Linguaggio_assembly&action=edit&redlink=1) si definiscono tre [locazioni di memoria](https://it.wikiversity.org/w/index.php?title=Locazioni_di_memoria&action=edit&redlink=1) che indicheremo con le lettere A, B, C. Quando il [programma](https://it.wikiversity.org/w/index.php?title=Programma&action=edit&redlink=1) parte esso carica nella [CPU](https://it.wikiversity.org/wiki/CPU) il numero presente in A e lo somma con quello che si trova nella posizione B memorizzando il risultato in C.
* In un [linguaggio BASIC](https://it.wikiversity.org/w/index.php?title=Linguaggio_BASIC&action=edit&redlink=1) il programma da semplicemente la direttiva che il numero nella [locazione di memoria](https://it.wikiversity.org/w/index.php?title=Locazione_di_memoria&action=edit&redlink=1) C deve essere uguale alla somma dei numeri nelle posizioni A e B.

# Programmi Decisionali

I programmi informatici risultano essere molto più articolati di un processo lineare come l'addizione di due numeri. Per dare al programma una struttura più complessa si utilizzano i seguenti strumenti:

* Le **diramazioni** consistono nel fornire al codice delle funzioni che possono cambiare lo svolgimento dell'[algoritmo](https://it.wikiversity.org/w/index.php?title=Algoritmo&action=edit&redlink=1) a seconda del risultato della condizione espressa.
* I **cicli** sono sequenze di istruzioni eseguite ripetutamente fin quando non viene soddisfatta una determinata condizione. Ogni ripetizione è detta iterazione.

# Le Tecniche di Programmazione

Esistono varie tecniche di programmazione, le più comuni ed utilizzate sono le seguenti:

* [Programmazione non strutturata](https://it.wikiversity.org/w/index.php?title=Programmazione_non_strutturata&action=edit&redlink=1) - in cui il programma è costituito da un unico blocco di istruzioni (detto *main*) che vengono eseguite in maniera sequenziale.
* [Programmazione procedurale](https://it.wikiversity.org/w/index.php?title=Programmazione_procedurale&action=edit&redlink=1) - in cui vi sono blocchi di istruzioni che fungono da sottoprogrammi e svolgono funzioni ben definite.
* [Programmazione modulare](https://it.wikiversity.org/w/index.php?title=Programmazione_modulare&action=edit&redlink=1) - in cui è possibile utilizzare procedure messe a disposizione da altri programmi e permettere a questi di poter utilizzare le proprie.
* [Programmazione orientata agli oggetti](https://it.wikiversity.org/wiki/Programmazione_orientata_agli_oggetti) - in cui vengono creati degli "oggetti" che possono funzionare autonomamente o interagire tra di loro. Ogni oggetto possiede in sé delle specifiche proprietà e funzioni (se ipotizziamo come oggetto un'automobile, una caratteristica sarà il suo colore o la sua marca e le funzioni saranno l'accelerare o il frenare).