



IVANA SACCHI

# ATTIVITÀ DI CODING NELLA SCUOLA PRIMARIA

*IL SOFTWARE ALLEGATO FUNZIONA CON  
TUTTI I SISTEMI OPERATIVI*



# SOMMARIO

41 Problemi liberi

43 Tartaruga

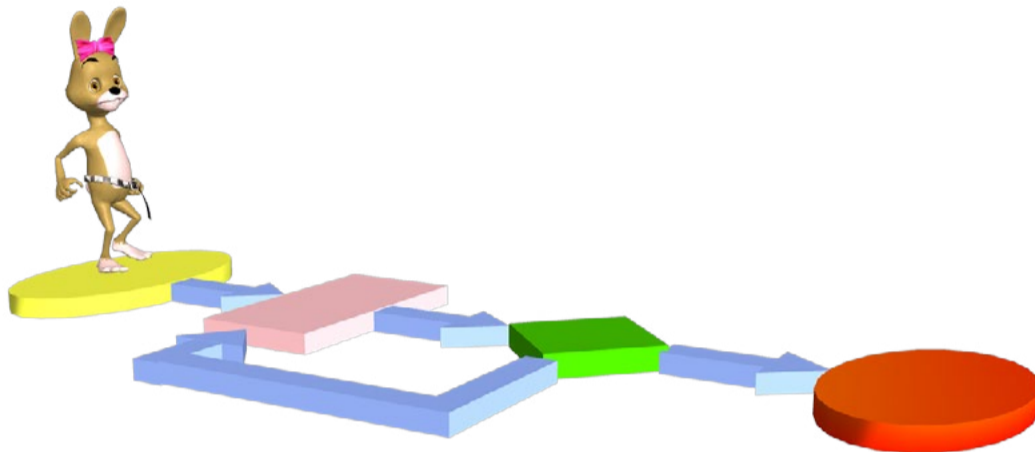
45 Indicazioni per il download del software

- 2 Coding - il pensiero computazionale
- 3 Blockly
- 4 Download del software
  - 4 Blockly su tablet
- 6 Utilizzare Blockly
- 9 Betta Coniglietta
  - 9 Betta Coniglietta - Trova le carote
  - 11 Quali carote?
  - 11 Massimo .... blocchi - 1°
  - 12 Riordina i blocchi - 1°
  - 12 Ottimizza il codice
  - 13 Tanti passi
  - 14 Massimo .... blocchi - 2°
  - 14 Riordina i blocchi - 2°
  - 15 Usa ... blocchi
  - 15 Betta torna a casa
  - 16 Betta sbaglia strada
- 17 Milly
- 18 Ranocchia
  - 18 Aiuta Ranocchia a raggiungere la ninfea
  - 19 Nuovi stagni
- 20 Ape Logica
  - 20 Ripeti fino al fiore
    - 22 Soluzioni
  - 24 Avanti e indietro
    - 24 Soluzioni
  - 26 Se c'è strada ...
    - 26 Soluzioni
- 28 Strade
  - 28 Conduci nel garage - 1°
    - 28 Soluzioni
  - 29 Conduci nel garage - 2°
    - 29 Soluzioni
  - 30 Conduci nel garage - 3°
    - 30 Soluzioni
- 32 Colora le collane
- 35 Missione spaziale
- 36 Problemi
  - 37 Problemi 1
  - 38 Problemi 2
  - 39 Problemi 3
  - 39 Problemi 4
  - 41 Problemi 5

# CODING - IL PENSIERO COMPUTAZIONALE

*Coding* è un termine inglese che indica la stesura di programmi informatici mediante un codice. Il suo significato letterale italiano è *programmare*.

Ma cosa vuol dire *programmare* in ambito informatico? Significa organizzare un *algoritmo* per trovare una soluzione ad un problema. L'algoritmo può essere pensato nel nostro linguaggio naturale, ma va poi sviluppato utilizzando le istruzioni presenti nel catalogo del linguaggio scelto. Un poco come quando utilizziamo un elaboratore di testi: possiamo pensare di voler evidenziare del testo con il grassetto, ma per farlo dobbiamo utilizzare il pulsante adatto, che il computer riconosce. Programmare significa quindi progettare un *algoritmo* e scriverlo con un codice che il computer sia in grado di riconoscere.



Quando si parla di *coding* a scuola l'argomento diventa più ampio. L'obiettivo non è la scrittura di codice (che diventa uno strumento), ma piuttosto l'acquisizione delle capacità che permettono di risolvere un problema pianificando una strategia: il *pensiero computazionale*. Viene definito come la capacità di scomporre un problema complesso in diverse parti, che risultano più gestibili se affrontate una per volta. Trovando una soluzione a ciascuna delle parti è possibile risolvere il problema generale.

Gli step per la programmazione, che diventano step didattici, sono quindi questi:

- Individuare i dati di un problema.
- Organizzare i dati in base a criteri logici.
- Ipotizzare la soluzione del problema.
- Progettare un algoritmo definendone i passaggi per la soluzione del problema.
- Organizzare l'algoritmo utilizzando le istruzioni corrette disponibili nel catalogo.
- Confrontare soluzioni possibili per identificare la soluzione migliore secondo un criterio.
- Generalizzare un processo di risoluzione del problema per poterlo trasferire ad altri problemi.

Il tutto può essere proposto a "misura di bambino", utilizzando strumenti e cataloghi di istruzioni che possano essere sperimentati come un gioco.

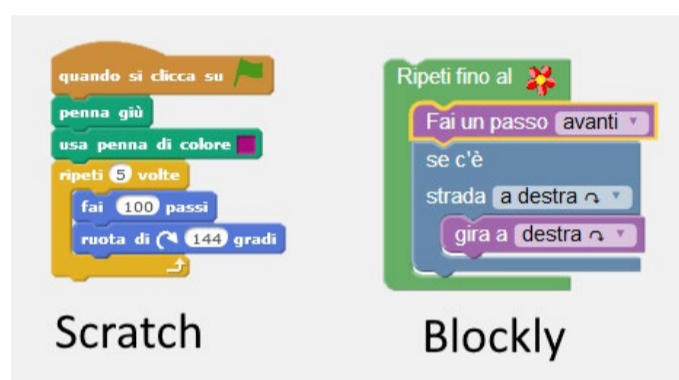
# BLOCKLY

*Blockly* è un ambiente strutturato che propone attività “*pronte per essere eseguite*”. In ogni attività sono presenti la definizione del problema in formato grafico, il catalogo di istruzioni per risolverlo e l’area di lavoro.

Può essere utilizzato on-line ([xxxxxxxxxxxxx](#)) oppure scaricato su computer o tablet. L’utilizzo off-line lo rende funzionale anche all’utilizzo domestico, qualora per motivi di sicurezza si preferisca che il bambino non acceda alla rete senza la presenza di adulti.

Le attività possono essere sperimentate coralmemente alla LIM, oppure affrontate individualmente individualmente e a piccoli gruppi. Nel testo sono presenti anche alcuni suggerimenti per l’utilizzo alla Lim, affiancando l’attività con rappresentazioni sul quaderno e con altre attività che non richiedono l’utilizzo del computer.

Blockly per la realizzazione del codice utilizza la stessa tipologia di blocchi di [Scratch](#), Può quindi essere considerato un’attività propedeutica all’utilizzo di questo ambiente “aperto”.



La stessa tipologia di blocchi “*a puzzle*” viene utilizzata da molti altri programmi, per cui ciò che si apprende è poi spendibile in molti ambienti di programmazione.

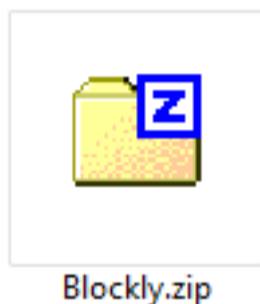
*Blockly* è un ambiente già ricco di attività, ma non va considerato terminato. Verrà aggiornato ed ampliato. Consiglio quindi di verificare sul sito [www.ivana.it](http://www.ivana.it) se vi sono aggiornamenti ed, in caso positivo, ri-scaricare il testo dalla libreria (il nuovo download del testo aggiornato è gratuito) e scaricare nuovamente il software.



## DOWNLOAD DEL SOFTWARE

Effettuare il download del software dal sito [www.ivana.it](http://www.ivana.it). Per l'operazione è necessario eseguire l'accesso all'area privata. Le credenziali sono riportate nell'ultimo capitolo (“*Indicazioni per il download del software*”).

Verrà scaricato il file *Blockly.zip*. È sufficiente decomprimerlo. L'immagine dell'icona potrebbe essere diversa da quella rappresentata nell'immagine sotto (l'immagine viene attribuita dal software installato sul computer). L'estensione *.zip* è generalmente nascosta.



Per decomprimere il file dovrebbe essere sufficiente un *clic destro sul file*. Nella finestra contestuale che viene visualizzata, scegliere la funzione *Estrai tutto*.

In alternativa è possibile utilizzare un file di decompressione, ad esempio 7-Zip, scaricabile all'indirizzo [www.7-zip.org](http://www.7-zip.org).

Al termine, si otterrà una cartella di nome *Blockly*: questa cartella contiene tutto il pacchetto del software.



## BLOCKLY SU TABLET

È possibile utilizzare Blockly anche su tablet. Per l'utilizzo on-line è sufficiente connettersi all'indirizzo [XXXXXXXXXX](http://XXXXXXXXXX), per l'utilizzo off-line consiglio di scaricare e decomprimere il pacchetto con un computer, come indicato sopra. Salvare poi la cartella decompressa sull'SD del tablet.

Per visualizzare i file HTML off-line è necessario un visualizzatore: in caso contrario verrebbero visualizzati come file di testo.

Per il sistema *Android* ve ne sono molti disponibili sul *Play Store*. Ho testato il funzionamento con [HTML Reader/ Viewer](#) e ho verificato il corretto funzionamento, anche se lo spazio di lavoro risulta ovviamente più “ristretto” rispetto all’utilizzo su computer. Utilizzandolo in verticale i blocchi del catalogo talvolta non sono visibili, in quanto lo spazio non è sufficiente per mostrare contemporaneamente area di lavoro e catalogo. Cliccare sulla parola *Catalogo* per visualizzare i blocchi a disposizione e trascinare nell’area di lavoro quelli necessari.



Non sono in grado di verificare il funzionamento con visualizzatori su sistema *iOS*®, ma ipotizzo non vi siano particolari differenze.

# UTILIZZARE BLOCKLY

Tutte le attività sono contenute all'interno della cartella *Blockly*, ottenuta dopo la decompressione del file scaricato. Aprire la cartella e avviare il file *Index.html* (l'estensione *.html* potrebbe non essere visibile).

Non sono necessarie installazioni. *Blockly* viene avviato tramite il browser predefinito sul computer. Non è necessaria connessione in rete.



Al primo avvio è possibile che si presenti una finestra nella quale viene richiesto quale browser utilizzare: scegliere il browser e selezionare "Utilizza sempre questo programma".

Qualora il file *Index* non si avvia con il browser, effettuare un clic destro sul file; nel menù contestuale che viene visualizzato scegliere "Apri con..." e scegliere un browser tra quelli disponibili sul computer.

*Blockly* funziona correttamente con *Google Chrome*, *Firefox Mozilla*, *Safari* e *Microsoft Edge*.

Vi potrebbero essere invece malfunzionamenti utilizzando *Internet Explorer*. Se sul computer è presente solo questo browser, consiglio l'installazione di *Google Chrome* oppure *Firefox Mozilla*,




In *Blockly* sono presenti vari ambienti, che focalizzano problemi di tipo diverso e quindi utilizzano cataloghi di istruzioni diverse.

In ognuna di queste aree sono presenti più attività. Alcune aree (ad esempio *Betta Coniglietta* e *Problemi*) sono organizzate in sotto-aree. Per ogni area (o sotto-area) sono presenti numerose proposte.

Avviando un'attività viene visualizzato il menù di navigazione.



Le immagini possono cambiare, ma la struttura di navigazione rimane sempre la stessa:

	L'immagine di <i>Blockly</i> è sempre presente e riporta all'indice generale (la pagina di avvio).
	La seconda immagine è sempre presente, ma varia per ogni area; porta all'indice dell'area.
	La terza immagine è presente solo se vi è una sotto-area: porta all'indice della sotto-area.

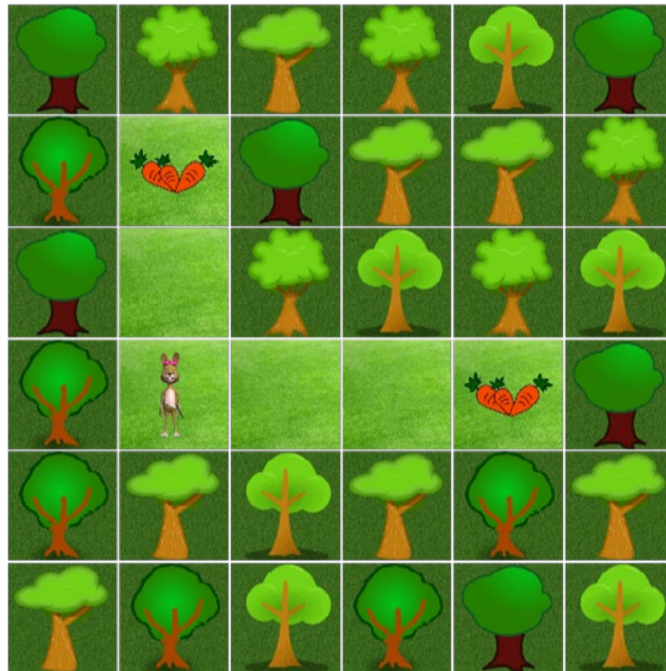
	<p>I pulsanti numerati permettono di navigare liberamente tra gli esercizi proposti nell'attività aperta. Generalmente vi sono 12 esercizi. Il colore giallo indica l'esercizio corrente, il colore verde quelli completati; alla chiusura del browser lo storico viene perso, quindi tutti i pulsanti alla riapertura saranno bianchi.</p>
--	---

Come evidenziato nella presentazione del menù, in ogni attività vi sono più proposte (generalmente 12). Le 12 attività sono graficamente diverse, ma tutte presentano lo stesso problema. Non è quindi necessario che vengano svolte tutte. Se il bambino completa con facilità due-tre proposte, consiglio di procedere con l'attività successiva. La numerosità è finalizzata al permettere al bambino, qualora avesse bisogno di aiuto all'inizio, di mettersi in gioco poi con problemi simili e generalizzare le strategie sperimentate.

Durante l'attività, è importante non sottolineare gli errori, ma al contrario, valorizzare il metodo sperimentale: il bambino ipotizza la strategia, compone con i blocchi la lista di istruzioni, controlla il funzionamento ed interviene per modificare gli eventuali errori.

Nella pagina è poi presente l'area del problema: nell'esempio a fianco *Betta Coniglietta* deve scegliere la strada per raggiungere e mangiare le carote.

In ogni area vi sono problemi di tipo diverso, ma è generalmente molto facile comprendere la richiesta. Il processo mentale viene attivato nell'analizzare il catalogo e trovare strategie di soluzione.



Il catalogo fornisce i blocchi utilizzabili per progettare e costruire l'algoritmo risolutivo. Ogni area dispone di cataloghi funzionali diversi, anche se alcuni blocchi possono essere presenti in più aree.

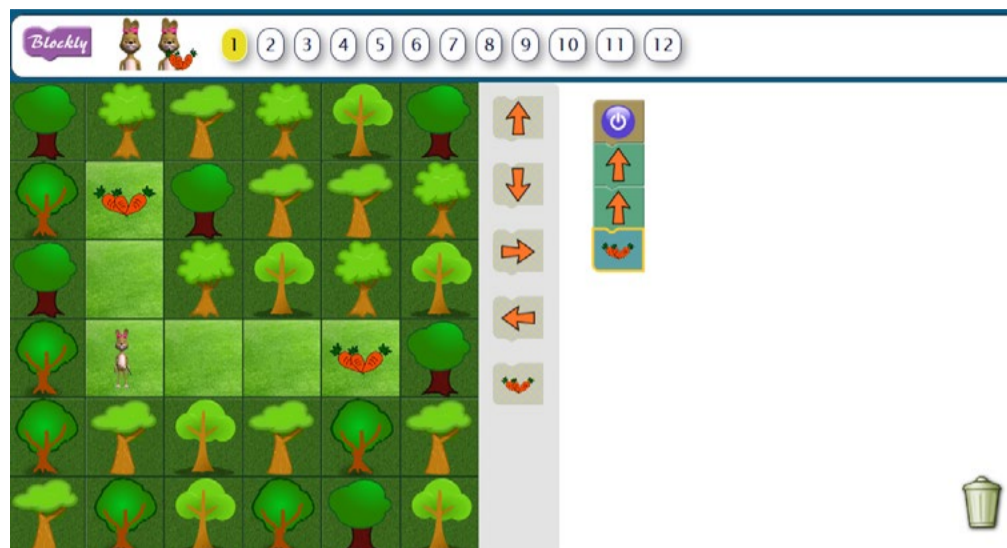





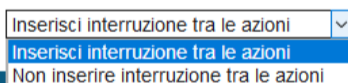

I blocchi vanno composti e agganciati tra di loro per realizzare l'algoritmo risolutivo. In alcune attività è previsto l'utilizzo di un numero massimo di blocchi. Raggiunto questo numero, i blocchi nel catalogo risultano disabilitati (come nell'esempio).



Per agganciare due blocchi è sufficiente avvicinarli trascinando un blocco vicino a quello al quale deve essere agganciato: nelle prime attività i blocchi mostrano il disegno dell'aggancio sopra e sotto. Il primo blocco (che è già posizionato) non ha l'aggancio sopra e l'ultimo (mangiare le carote) non ha l'aggancio sotto.

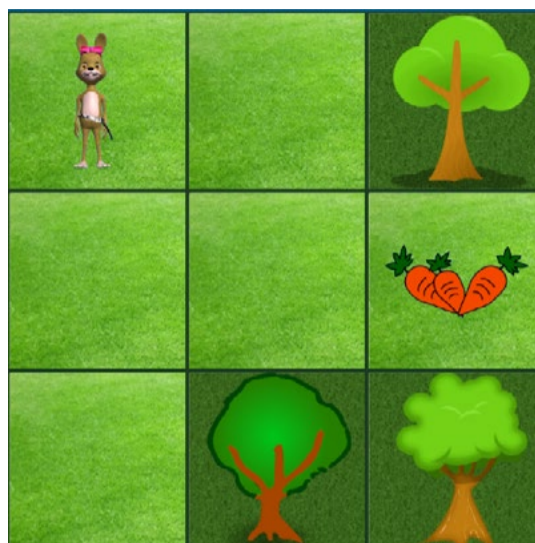
Per sganciare due blocchi, cliccare e trascinare quello in basso. I blocchi inseriti per errore possono essere eliminati trascinandoli nel cestino, oppure nell'area del *Catalogo*.



	<p>Durante il posizionamento e l'aggancio dei blocchi non viene visualizzata alcuna animazione. Per avviare l'animazione premere il pulsante <i>Avvia</i>, sotto l'area del problema.</p> <p>Viene avviata l'animazione corrispondente alla serie di istruzioni inserite. Durante l'animazione, il blocco corrispondente all'istruzione corrente "si illumina", in modo che sia possibile visualizzare il processo mantenendo sincronizzati codice e animazione.</p>
	<p>Il pulsante <i>Ricomincia</i> riporta l'animazione alla situazione iniziale, quindi permette di verificare nuovamente l'animazione, dopo eventuale modifica del codice.</p>
	<p>In alcune attività è presente lo slider che consente di impostare la velocità di riproduzione dell'animazione; impostando una bassa velocità è più facile l'operazione di debugging (la ricerca degli errori); una maggiore velocità può essere utile quando ci si sente sicuri dell'algoritmo progettato.</p>
	<p>In alcune aree è anche possibile optare o meno per l'inserimento di una breve interruzione tra l'esecuzione di un'istruzione e la successiva. Anche questo può essere utile per la ricerca di errori di progettazione.</p>
	<p>Il pulsante <i>Salva Disegno</i> permette di salvare in formato immagine l'area del problema. Il file immagine viene realizzato tra i file temporanei ed aperto nell'editor di immagini predefinito sul computer. Utilizzare la funzione <i>Salva con nome</i> dell'editor se si desidera salvare il file. Questo pulsante potrebbe risultare non funzionante in alcuni browser.</p>

# BETTA CONIGLIETTA

L'area di Betta è la più semplice.



Aperto le attività, a sinistra viene visualizzato il problema: Betta Coniglietta deve mangiare le carote. In alcune attività successive dovrà invece tornare a casa.

Il catalogo di istruzioni è molto semplice: un passo (oppure più passi) in ogni direzione, più l'istruzione per mangiare le carote.

Per quanto semplice, l'attività richiede di selezionare le istruzioni adatte e ordinarle: la realizzazione dell'algoritmo.

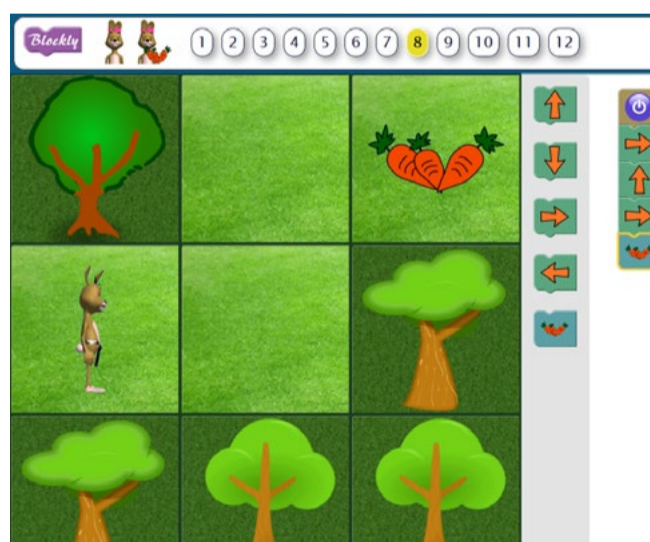
Trascinare i blocchi dal catalogo nell'area di lavoro ed agganciarli per realizzare l'algoritmo.

Quando la serie di istruzioni è completa cliccare sul pulsante blu (quello in basso) in modo che il codice venga eseguito.

Mentre Betta esegue le istruzioni, il blocco corrispondente all'istruzione corrente "si illumina", in modo che sia possibile visualizzare il processo.

Lo slider con le immagini della chiocciola e della lepre permette di rendere più lenta o veloce l'animazione.

## BETTA CONIGLIETTA - TROVA LE CAROTE



Il compito è aiutare Betta Coniglietta a mangiare le carote. Nelle varie attività le strade possono diventare più o meno complesse, ma le attività dovrebbero risultare comunque abbastanza semplici: questo non esclude errori di progettazione; questi sono inevitabili. È importante che il bambino impari ad avviare l'animazione per verificare la propria lista di istruzioni e, se necessario, modificarla. Può anche essere utile imparare ad avviare l'animazione anche se il codice non è ancora completo, in modo di verificarne alcuni passaggi.