



# Documento di progettazione e condivisione dell'attività didattica



**Boosting** Digital  
Capacity





# Le avventure del coding

## Descrizione attività

Questa Unità di Apprendimento introduce gli studenti ai concetti base del coding e del pensiero computazionale attraverso attività ludiche e pratiche, sia unplugged che con semplici strumenti digitali. Verranno esplorati concetti come sequenze, istruzioni, cicli e debugging, stimolando la logica e la risoluzione dei problemi.

Destinatari: classe seconda e terza scuola primaria

### Discipline coinvolte:

- Tecnologia
- Italiano
- Geografia
- Matematica

Attività progettata nell'a.s. 2024-2025 da:

Direzione Didattica Collodi Torino

insegnanti : Ansini



# Obiettivi



## Obiettivi di apprendimento

### Al termine dell'attività, gli studenti **saranno in grado di:**

- Riconoscere e identificare sequenze di istruzioni.
- Spiegare e interpretare semplici algoritmi.
- Applicare ed eseguire istruzioni per raggiungere un obiettivo.
- Analizzare e organizzare una serie di passaggi per risolvere un problema.
- Progettare, creare e costruire percorsi utilizzando il pensiero computazionale.

### Obiettivi DigComp

- **1. Alfabetizzazione su informazioni e dati:**
  - 1.1 Navigare, ricercare e filtrare dati, informazioni e contenuti digitali (es. per trovare blocchi di codice specifici in un ambiente visuale).
- **2. Comunicazione e collaborazione:**
  - 2.1 Interagire con gli altri attraverso le tecnologie (es. collaborare a coppie o in gruppo per risolvere sfide di coding).
- **3. Creazione di contenuti digitali:**
  - 3.1 Sviluppare contenuti digitali (es. creare semplici sequenze di codice per animare personaggi).
  - 3.4 Programmazione (obiettivo primario dell'UdA).
- **5. Risolvere problemi:**
  - 5.1 Risolvere problemi tecnici (es. debuggare il proprio codice).
  - 5.3 Utilizzare in modo creativo le tecnologie digitali (es. trovare soluzioni originali a problemi di coding).

### Al termine dell'attività gli studenti **conosceranno:**

- Il concetto di sequenza e istruzione.
- L'importanza dell'ordine nelle istruzioni.
- Cosa sia un "bug" (errore) e come correggerlo (debugging).
- Esempi pratici di come il coding è presente nella vita quotidiana.

# Posizione nel DigComp



## Alfabetizzazione su informazioni e dati

- 1.1. Navigare, ricercare e filtrare dati, informazioni e contenuti digitali
- 1.2. Valutare dati, informazioni e contenuti digitali
- 1.3. Gestire dati, informazioni e contenuti digitali



## Comunicazione e collaborazione

- 2.1. Interagire con gli altri attraverso le tecnologie
- 2.2. Condividere informazioni attraverso le tecnologie digitali
- 2.3. Esercitare la cittadinanza attraverso le tecnologie digitali
- 2.4. Collaborare attraverso le tecnologie digitali
- 2.5. Netiquette
- 2.6. Gestire l'identità digitale



## Creazione di contenuti digitali

- 3.1. Sviluppare contenuti digitali
- 3.2. Integrare e rielaborare contenuti digitali
- 3.3. Copyright e licenze
- 3.4. Programmazione



## Sicurezza

- 4.1. Proteggere i dispositivi
- 4.2. Proteggere i dati personali e la privacy
- 4.3. Proteggere la salute e il benessere
- 4.4. Proteggere l'ambiente



## Risolvere problemi

- 5.1. Risolvere problemi tecnici
- 5.2. Individuare bisogni e risposte tecnologiche
- 5.3. Utilizzare in modo creativo le tecnologie digitali
- 5.4. Individuare i divari di competenze digitali

posiziona il segnalino verde!



# ANALISI SWOT

## PUNTI DI FORZA

- L'approccio ludico e pratico che cattura l'attenzione dei bambini.
- Stimolo del pensiero logico e della creatività.
- Promozione del lavoro di squadra e della collaborazione.
- Sviluppo competenze trasversali utili in diverse discipline.

## STRENGTHS

## PUNTI DI DEBOLEZZA

- Potenziale mancanza di dispositivi per ogni studente.
- Necessità di un'adeguata formazione per gli insegnanti.

## WEAKNESSES

## SWOT

## OPPORTUNITIES

## OPPORTUNITÀ

- Disponibilità di risorse online gratuite per il coding per bambini (es. Code.org, ScratchJr).
- Possibilità di integrare il coding con altre materie curriculari.
- Sviluppo di competenze digitali fondamentali per il futuro.

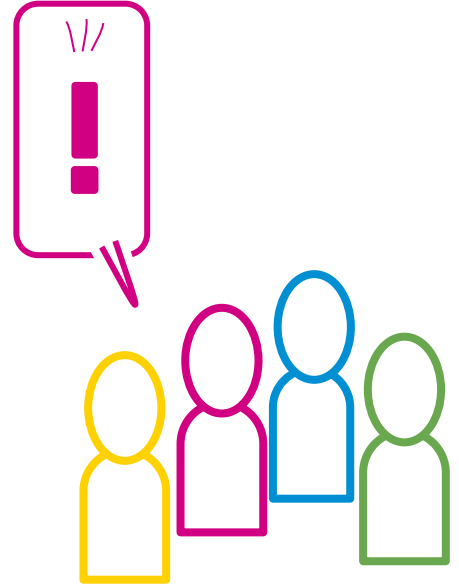
## THREATS

## MINACCE

- Possibili problemi con connessione Internet



# FEEDBACK



Empty rounded rectangular box for notes or feedback.

Empty rounded rectangular box for notes or feedback.

# Metodologie e valutazione



## Metodologie didattiche

Seleziona le **metodologie** più adatte per raggiungere gli obiettivi di apprendimento:

- |   |  |
|---|--|
| <input checked="" type="checkbox"/> Apprendimento cooperativo | <input type="checkbox"/> Peer tutoring                     |
| <input type="checkbox"/> Debate                               | <input checked="" type="checkbox"/> Problem-based learning |
| <input checked="" type="checkbox"/> Didattica laboratoriale   | <input checked="" type="checkbox"/> Project-based learning |
| <input checked="" type="checkbox"/> Gioco di ruolo            | <input type="checkbox"/> Altro: .....                      |
| <input type="checkbox"/> Lezione frontale                     |  |

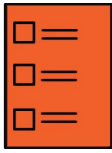
## Cosa valutare

### Per valutare l'attività **osserva**:

- La capacità di seguire e creare sequenze di istruzioni.
- La partecipazione attiva alle attività di gruppo.
- La logica dimostrata nella risoluzione dei "bug".
- La creatività nella progettazione di soluzioni.

### Puoi valutare l'attività **mediante**:

1. Osservazione diretta durante le attività pratiche.
2. Brevi questionari o schede di autovalutazione semplificate.
3. Valutazione dei "prodotti" realizzati (es. percorsi disegnati, animazioni semplici).
4. Checklist di verifica delle competenze acquisite.



# Preparazione

## Durata

6 ore

## Setting d'aula

- Tavoli ad isola (per facilitare il lavoro di gruppo e lo scambio di idee).
- Un'area dedicata ai materiali unplugged (scacchiera, carte, ecc).
- Un'area con i dispositivi tecnologici (tablet/computer, se disponibili).

## Tecnologie e strumenti

Elenca le tecnologie e gli strumenti necessari per l'attività. Esempio:

- scacchiera sul pavimento (realizzata anche con scotch di carta)
- carte direzionali
- cartoncini, pennarelli/matite, forbici
- Robot educativi programmabili (es. Bee-Bot, Doc)
- tablet/notebook

## Cosa è necessario fare prima dell'attività

Dettaglia qui la tua checklist. Esempio:

- Preparare il materiale "unplugged" (carte con frecce direzionali, griglie a scacchiera, disegni di personaggi e ostacoli).
- Preparare schede guida semplici per le attività.
- Predisporre esempi pratici o video introduttivi sul coding per bambini.
- Verificare la funzionalità dei dispositivi tecnologici (tablet/PC) e la connessione internet.
- Installare le app di coding visuale (es. ScratchJr) sui dispositivi.



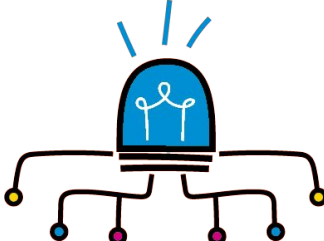
# Scaletta

Durata	Azioni docente e studente	Strumenti necessari
Es: 5min	<p>Racconta, passo dopo passo, come condurre l'attività didattica in classe, come traccia per l'erogazione e come stimolo per permettere ad altri di ripetere l'esperienza.</p> <p>Azioni dei partecipanti:</p> <p>Azioni del facilitatore:</p>	Esempio: Cartellone, Post-it, Pennarelli
<b>Caccia al tesoro con istruzioni (2 ore)</b>	<p><b>Preparazione:</b> Creare mappe semplici su fogli grandi o sul pavimento della classe con un punto di partenza e un "tesoro" da raggiungere. Posizionare ostacoli. Preparare carte con istruzioni direzionali (avanti, indietro, destra, sinistra, salta).</p> <p><b>Docente:</b> Introduce il concetto di "istruzione" e "sequenza". Spiega come le frecce indicano i movimenti. Divide la classe in piccoli gruppi. Assegna una mappa e un set di carte istruzione a ciascun gruppo. Spiega la regola di non toccare il "cody-robot" (un compagno o un pupazzo) ma solo di dargli le istruzioni.</p> <p><b>Studente:</b> A turno, in gruppo, gli studenti scelgono le carte istruzione per creare un percorso dal punto di partenza al tesoro. Un compagno può fare da "cody-robot" e seguire le istruzioni. Se il robot non raggiunge il tesoro, il gruppo deve "debuggare" (trovare e correggere) le istruzioni.</p>	Mappe disegnate, carte istruzione, pupazzi/compagni che fungono da robot.
<b>Caccia al tesoro con istruzioni (2 ore)</b>	<p><b>Preparazione:</b> Creare mappe semplici su fogli grandi o sul pavimento della classe con un punto di partenza e un "tesoro" da raggiungere. Posizionare ostacoli. Preparare carte con istruzioni direzionali (avanti, indietro, destra, sinistra, salta).</p> <p><b>Docente:</b> Introduce il concetto di "istruzione" e "sequenza". Spiega come le frecce indicano i movimenti. Divide la classe in piccoli gruppi. Assegna una mappa e un set di carte istruzione a ciascun gruppo. Spiega la regola di non toccare il "cody-robot" (un compagno o un pupazzo) ma solo di dargli le istruzioni.</p> <p><b>Studente:</b> A turno, in gruppo, gli studenti scelgono le carte istruzione per creare un percorso dal punto di partenza al tesoro. Un compagno può fare da "cody-robot" e seguire le istruzioni. Se il robot non raggiunge il tesoro, il gruppo deve "debuggare" (trovare e correggere) le istruzioni.</p>	Mappe disegnate, carte istruzione, pupazzi/compagni che fungono da robot.



# Scaletta

Durata	Azioni docente e studente	Strumenti necessari
<b>Crea il tuo ballo robotico (1.5 ore):</b>	<p><b>Preparazione:</b> Preparare delle carte con movimenti semplici (es. alza il braccio, fai un passo avanti, gira su te stesso, batti le mani).</p> <p><b>Docente:</b> Spiega che i robot eseguono solo le istruzioni che ricevono. Chiede ai bambini di creare una sequenza di movimenti per un "ballo robotico".</p> <p><b>Studente:</b> A coppie o in piccoli gruppi, i bambini scelgono le carte movimento e creano una sequenza per un breve ballo. Poi presentano il loro ballo alla classe, con un compagno che esegue i movimenti in base alle istruzioni scelte.</p>	Carte movimento, spazio libero in aula.
<b>Prime animazioni con ScratchJr (2 ore):</b>	<p><b>Preparazione:</b> Assicurarsi che ScratchJr sia installato sui tablet/computer. Preparare una breve introduzione all'interfaccia dell'app.</p> <p><b>Docente:</b> Introduce l'app ScratchJr, mostrando come selezionare i personaggi, gli sfondi e i blocchi di codice base (movimento, suono, aspetto). Propone una sfida: far muovere un personaggio da un punto all'altro dello schermo.</p> <p><b>Studente:</b> Gli studenti esplorano l'interfaccia di ScratchJr. Utilizzano i blocchi di codice per creare semplici sequenze di movimento per il loro personaggio. Sperimentano con diversi blocchi e osservano l'effetto. Provano a risolvere piccoli problemi se il personaggio non si muove come previsto.</p>	Tablet/Computer con ScratchJr installato, proiettore/LIM per la dimostrazione
<b>Cody e il labirinto (1.5 ore):</b>	<p><b>Preparazione:</b> Preparare semplici labirinti stampati o disegnati, o utilizzare piattaforme online come Code.org che offrono attività guidate.</p> <p><b>Docente:</b> Presenta l'idea di guidare un personaggio attraverso un labirinto usando le istruzioni di coding. Spiega il concetto di "ciclo" per ripetere le istruzioni.</p> <p><b>Studente:</b> Utilizzando un robot educativo (es. Bee-Bot) o un'attività online (Code.org), gli studenti programmano il personaggio per superare il labirinto. Devono ragionare sulle sequenze di istruzioni e individuare il percorso più efficiente.</p>	Labirinti (fisici o digitali), robot educativi (se disponibili), tablet/computer con accesso a Code.org o app simili.



# Condivisione di materiali prodotti

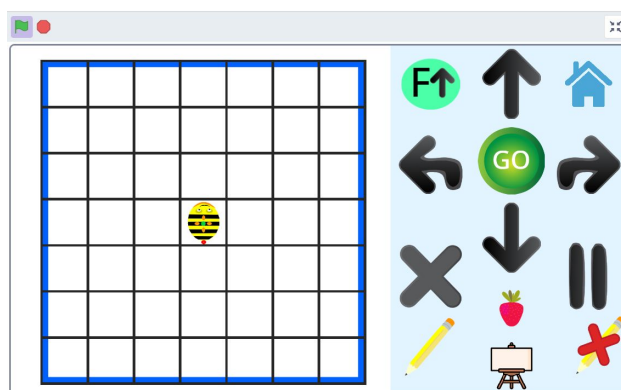
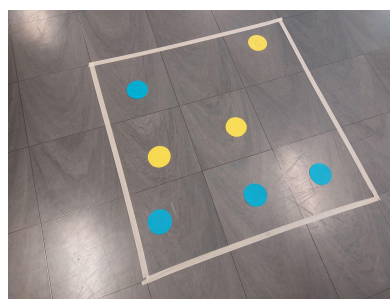
Puoi utilizzare questa slide (o crearne una ad hoc) per valorizzare e condividere al meglio l'esperienza inserendo foto, testi, screenshot, immagini, risorse utili.

Esempio:

- griglie di valutazione o questionari di gradimento utilizzati
- raccolta di lavori prodotti dagli studenti
- foto significative
- ...

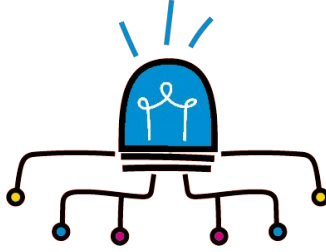


Esempio di attività su scacchiera



Simulatore bee-bot

Attenzione: non è possibile caricare immagini che raffigurino persone riconoscibili, i contenuti che non rispettano questa condizione verranno rimossi.



## Consigli e note

Scrivi qui i consigli che daresti ai tuoi colleghi. Puoi anche indicare idee, libri, siti web e app che consiglieresti per rendere l'attività incredibile.

- Iniziare sempre con attività "unplugged" per costruire le basi concettuali prima di passare al digitale.
- Incoraggiare il lavoro di gruppo e il "peer tutoring" per favorire lo scambio e l'aiuto reciproco.
- Celebrare ogni piccolo successo e incoraggiare la sperimentazione e l'errore come opportunità di apprendimento ("il bug è un amico!").
- Utilizzare un linguaggio semplice e figurato per spiegare i concetti di coding.
- Lasciare molto spazio alla creatività dei bambini.
  
- **Siti web:** Code.org (corsi gratuiti e strutturati), CoderDojo Italia (risorse e community).
- **App:** ScratchJr (per Android/iOS, facile e intuitiva), Lightbot (per risolvere puzzle con il coding), Kodable (per imparare le basi della programmazione).
- **Materiali:** Robot educativi come Bee-Bot, Doc o Lego WeDo.



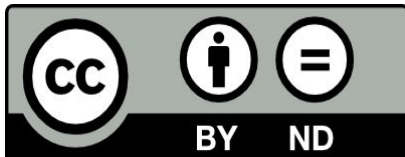
POST-IT

# Licenze

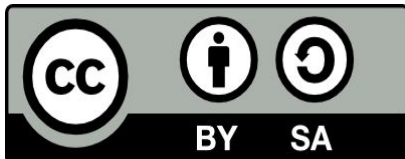
Scegli una licenza creative commons da attribuire a questo tuo documento  
Scopri [qui](#) cosa sono le licenze Creative Commons



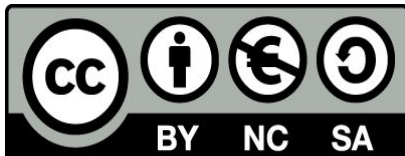
CC BY  
Attribuzione



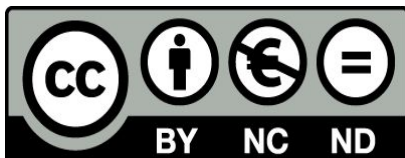
CC BY-ND  
Attribuzione - Non Opere Derivate



CC BY-SA  
Attribuzione - Condividi allo Stesso Modo



CC BY-NC-SA  
Attribuzione - Non Commerciale -  
Condividi allo Stesso Modo



CC BY-NC-ND  
Attribuzione - Non Commerciale - Non  
Opere Derivate

**CC BY-NC-SA**