Dall’uso critico alla AI Literacy:

Python e micro:bit come palestra di pensiero computazionale e consapevolezza sull’IA

[1] Giuliana Gelsomino e [2] Barbara Baldi

[1] *IC Virgilio-Gonzaga, Eboli (SA)*

[2] *IC Santorre di Santarosa, Savigliano (CN)*

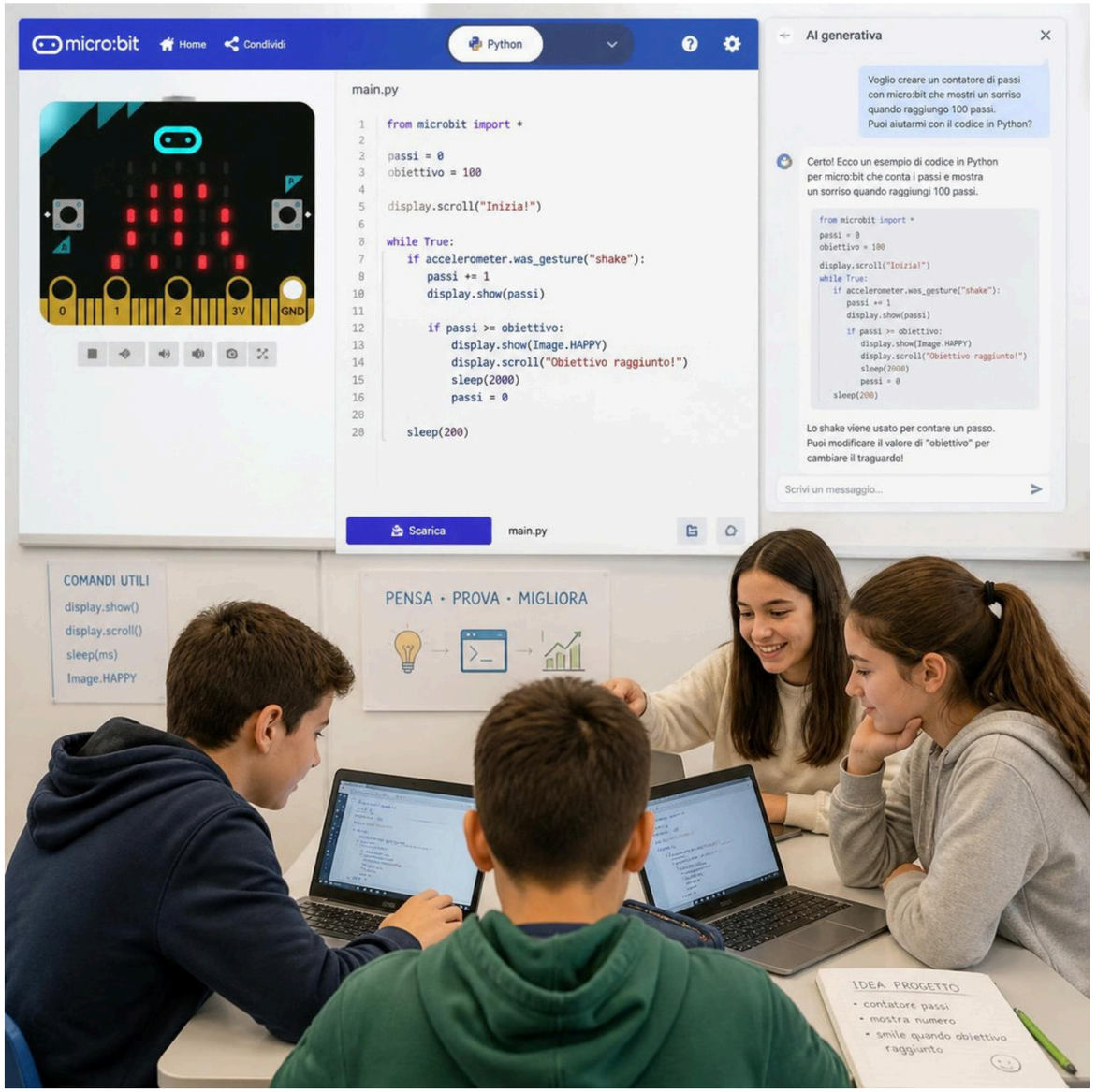
[ITADINFO]

CONVEGNO ITALIANO
SULLA DIDATTICA DELL’INFORMATICA

Padova, 9-11 ottobre 2026

- Usare IA generativa come supporto didattico per avviare alla programmazione testuale.
- Interaragire con IA come occasione di compresione attiva, riflessione metacognitiva e alfabetizzazione sull’IA.





Un nuovo scenario educativo

UNESCO Framework (2024)

Competenze e valori per formare co-creatori responsabili di tecnologia.

AI Act, art. 4 e Digital Omnibus (2026)

Alfabetizzazione obbligatoria sull'IA per un utilizzo consapevole dei rischi.

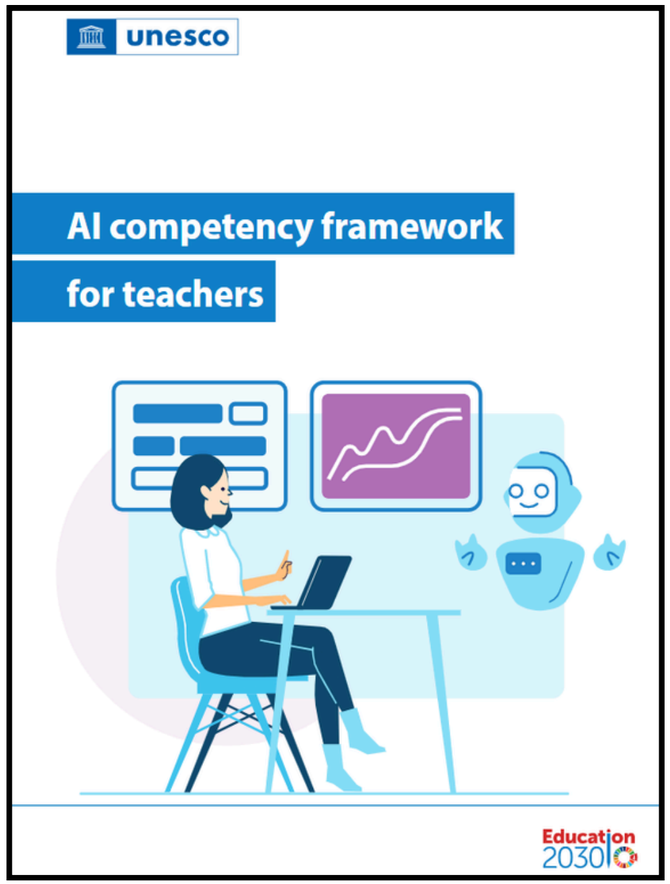
Indicazioni Nazionali 2025

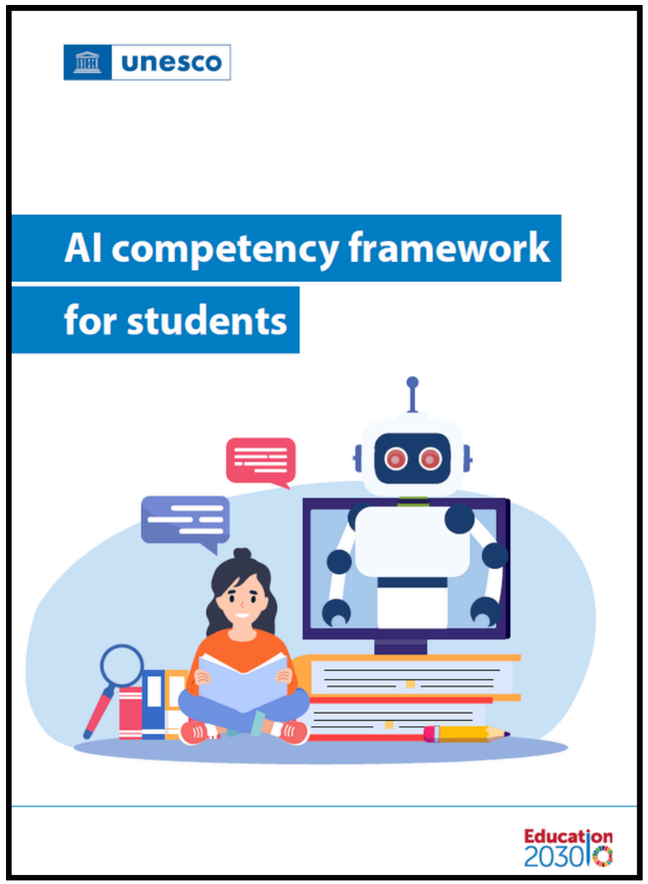
Valutazione critica dei sistemi di IA fin dalla scuola secondaria di I grado.

DigComp 3.0

La competenza sull’IA generativa entra in tutte le aree del framework

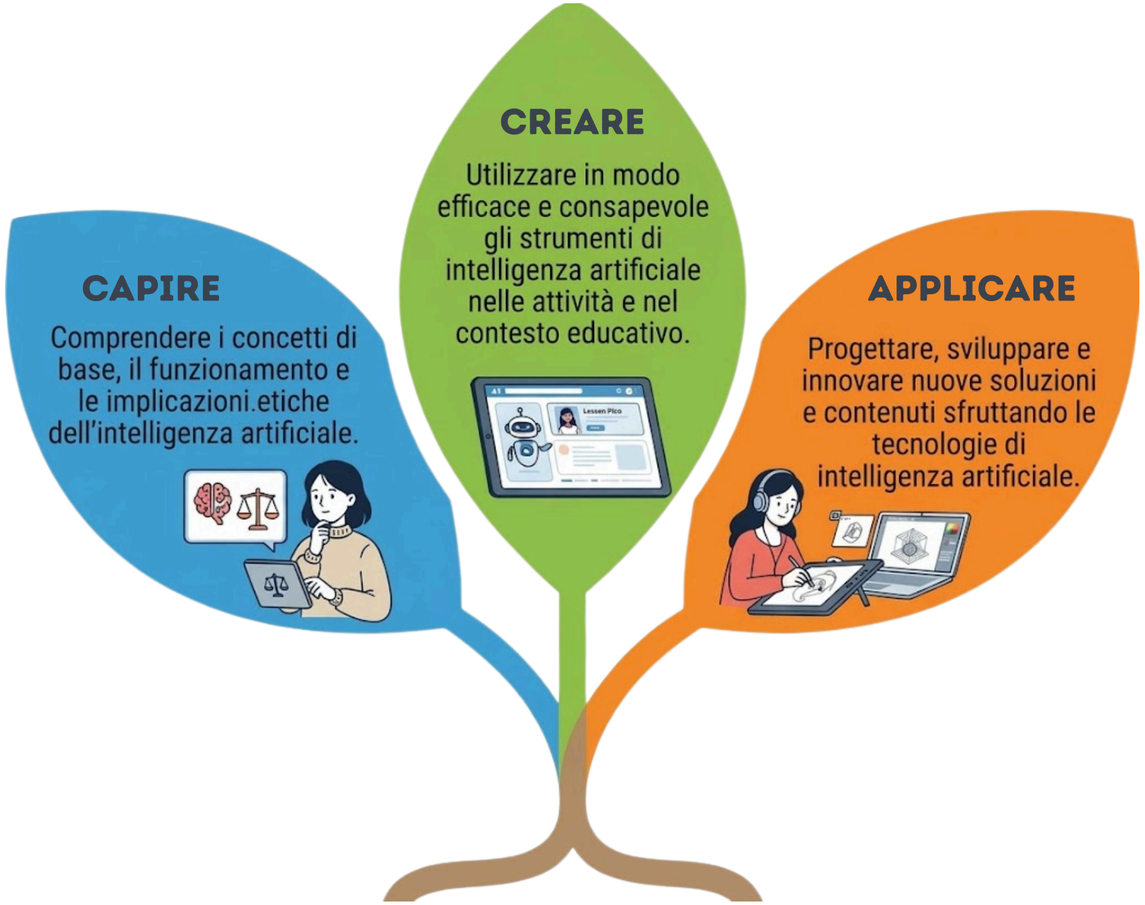
UNESCO e OECD-UE Framework

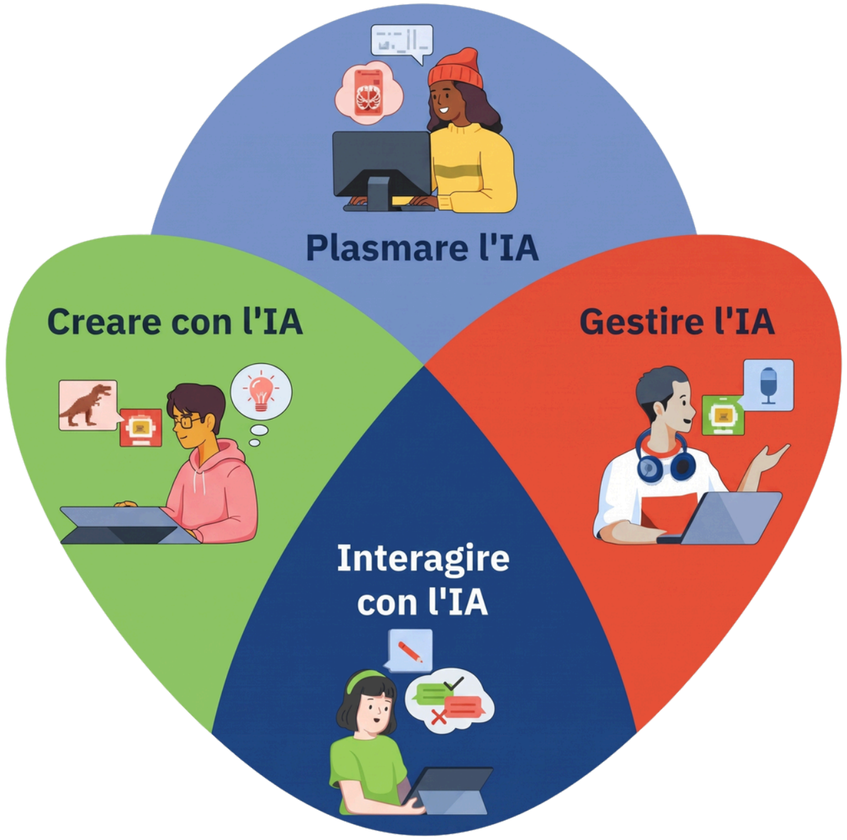






UNESCO Framework 2024/AILit framework OECD-UE 2026





UMC - PRIMM e Piramide di Anderson-Krathwohl

Esegue il codice per osservare il **comportamento effettivo** o testare l'ipotesi.

Confronta i risultati attesi ed effettivi per **convalidare o confutare** l'ipotesi iniziale.

Analizza il codice per **anticiparne** il comportamento e **formulare un'ipotesi.**

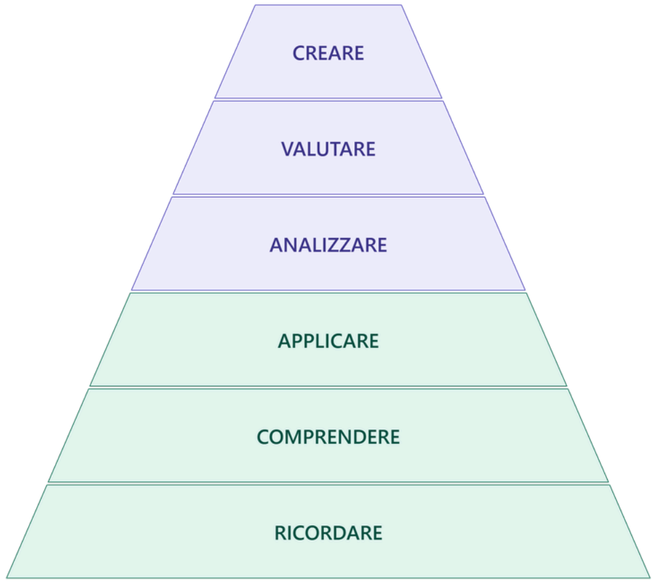
PREDICT (Analizzare)

RUN (Applicare)

INVESTIGATE (Valutare)

MODIFY (Analizzare/Creare)

MAKE (Creare)



USE (Applicare)

Esegue e interagisce con un programma o un gioco già **interamente sviluppato dall'insegnante.**

MODIFY (Analizzare/Creare)

Interviene sul codice esistente **modificando piccoli parametri, aggiungendo blocchi o risolvendo bug** (debugging).

CREATE (Creare)

Sviluppa un progetto o un programma del tutto **nuovo applicando in autonomia** i costrutti logici appresi.

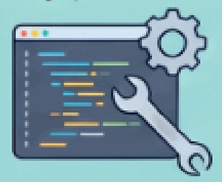
UMC, PRIMM e IA generativa

Modello UMC

Use (Usare)



Modify (Modificare)



Create (Creare)



Il Framework PRIMM affina il modello UMC

PRIMM rafforza la comprensione del codice prima di passare alla modifica e creazione.

Modello PRIMM

Predict, Run, Investigate

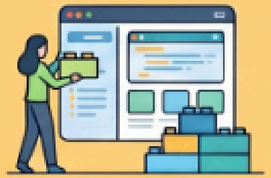


Predict Run Investigate

Modify



Make



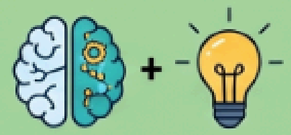
Tassonomia di Anderson & Krathwohl

Analizzare, Applicare, Valutare



Analizzare Applicare Valutare

Analizzare + Creare



Creare

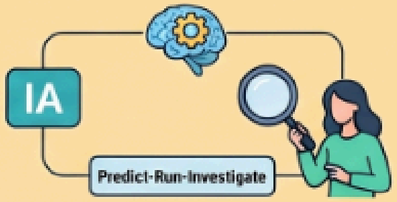


Integrazione dell'IA Generativa nell'Apprendimento



Prompting come Atto di Creazione

Formulare specifiche chiare per guidare l'IA richiede capacità di progettazione (Livello: Creare).



Revisione Critica tramite l'Indagine

Ripercorre Predict-Run-Investigate per valutare correttezza e qualità dell'output (Livello: Valutare).

Gemini Notebook

Metodologia: il ciclo di interazione con l’IA

1

PROMPTING Understanding AI / USE (PREDICT)



Metodologia: il ciclo di interazione con l’IA

1

PROMPTING Understanding AI / USE (PREDICT)



Comprensione concettuale e scomposizione

Comprendere il funzionamento dell’IA generativa e formulare consegne in linguaggio naturale.

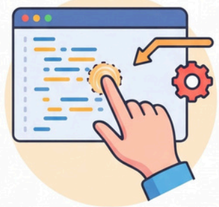
Collegamento: Fase 1 (Prompting)



Analisi preventiva del codice generato dall’IA prima della sua esecuzione nel simulatore.

2

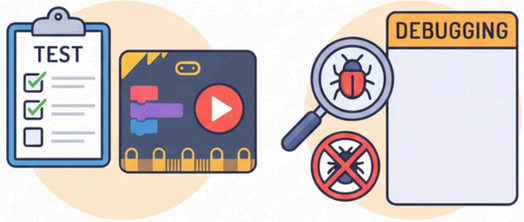
TEST USE (RUN + INVESTIGATE)→ MODIFY



Test, modifica e supervisione umana

Intervenire sul codice per correggere errori mantenendo il controllo diretto del processo.

Collegamento: Fase 2 (Test) e Fase 3 (Debugging)



Validazione su MakeCode, taratura dei parametri e formulazione di domande analitiche sui bug.

Metodologia: il ciclo di interazione con l’IA

1

PROMPTING Understanding AI / USE (PREDICT)



Comprensione concettuale e scomposizione

Comprendere il funzionamento dell’IA generativa e formulare consegne in linguaggio naturale.

Collegamento: Fase 1 (Prompting)



Analisi preventiva del codice generato dall’IA prima della sua esecuzione nel simulatore.

2

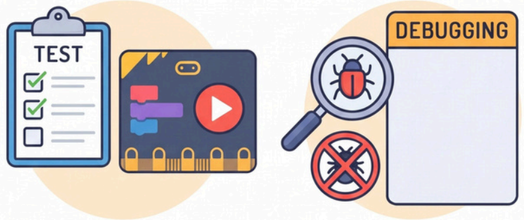
TEST USE (RUN + INVESTIGATE)→ MODIFY



Test, modifica e supervisione umana

Intervenire sul codice per correggere errori mantenendo il controllo diretto del processo.

Collegamento: Fase 2 (Test) e Fase 3 (Debugging)



Validazione su MakeCode, taratura dei parametri e formulazione di domande analitiche sui bug.

DEBUGGING Applying AI / MODIFY

3

Metodologia: il ciclo di interazione con l'IA

1 **PROMPTING** Understanding AI / USE (PREDICT)



Comprensione concettuale e scomposizione

Comprendere il funzionamento dell'IA generativa e formulare consegne in linguaggio naturale.

Collegamento: Fase 1 (Prompting)



Analisi preventiva del codice generato dall'IA prima della sua esecuzione nel simulatore.

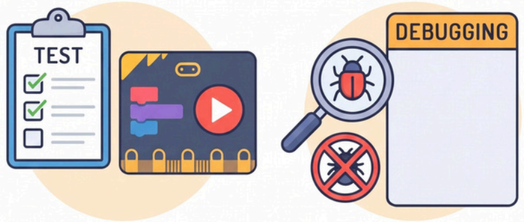
2 **TEST** USE (RUN + INVESTIGATE)→ MODIFY



Test, modifica e supervisione umana

Intervenire sul codice per correggere errori mantenendo il controllo diretto del processo.

Collegamento: Fase 2 (Test) e Fase 3 (Debugging)



Validazione su MakeCode, taratura dei parametri e formulazione di domande analitiche sui bug.

3 **DEBUGGING** Applying AI / MODIFY

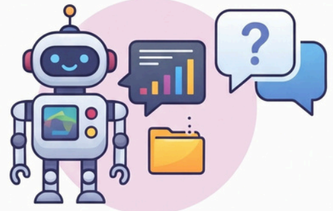
4 **SINTESI** Verso Creating with AI / CREATE



Co-creazione autonoma e valutazione etica

Usare l'IA come consulente per progetti complessi, riflettendo sull'impatto delle tecnologie.

Collegamento: Fase 4 (Sintesi e Creazione)



Sviluppo autonomo di artefatti originali e discussione sui limiti e responsabilità dell'IA.

Metodologia: il ciclo di interazione con l’IA



Comprensione concettuale e scomposizione

Comprendere il funzionamento dell’IA generativa e formulare consegne in linguaggio naturale.

Collegamento: Fase 1 (Prompting)



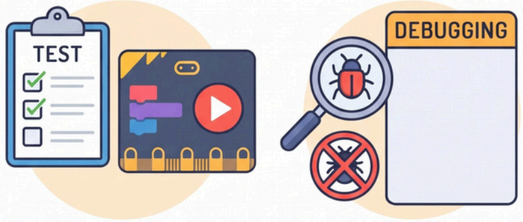
Analisi preventiva del codice generato dall’IA prima della sua esecuzione nel simulatore.



Test, modifica e supervisione umana

Intervenire sul codice per correggere errori mantenendo il controllo diretto del processo.

Collegamento: Fase 2 (Test) e Fase 3 (Debugging)



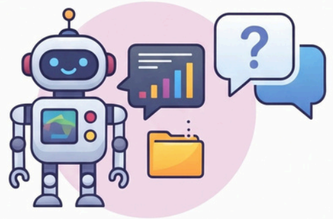
Validazione su MakeCode, taratura dei parametri e formulazione di domande analitiche sui bug.



Co-creazione autonoma e valutazione etica

Usare l'IA come consulente per progetti complessi, riflettendo sull'impatto delle tecnologie.

Collegamento: Fase 4 (Sintesi e Creazione)



Sviluppo autonomo di artefatti originali e discussione sui limiti e responsabilità dell'IA.

Perché questo ciclo funziona

Il codice generato dall’IA diventa un artefatto da leggere, testare, modificare e discutere.

La supervisione umana resta nel processo: l’IA non viene usata per “dare la soluzione”, ma per sostenere comprensione e problem solving.

Esempio di assistente

T

Tutor micro:bit Python

Assistente didattico per la scuola secondaria di primo grado che guida studenti e studentesse nella scrittura di codice Python per micro:bit. Non fornisce mai il codice risolutivo: spiega le funzioni già ...

Mostra altro



SCAN ME

Attività 1 – Primi passi

attività unplugged - Use

Attività 1 – Primi passi

Attività 1

"Hello, Micro:bit!"

Prompt: Cuore pulsante all'avvio della scheda.

Focus: Lettura guidata del codice Python prima dell'esecuzione.

AI Literacy: Comprensione preventiva (Understanding AI)

PROMPT SUGGERITO

Genera per me un semplice script in Python per l'editor Microsoft MakeCode per la scheda micro:bit (non MicroPython) che faccia lampeggiare un'icona a forma di cuore sul display LED, alternando un cuore grande e un cuore piccolo ogni mezzo secondo, in un ciclo infinito. Usa solo i comandi della categoria Basic. Non aggiungere commenti superflui: il codice deve essere semplice da leggere riga per riga.

Attività 1 – Primi passi

Use (predict)

Attività 1

"Hello, Micro:bit!"

Prompt: Cuore pulsante all'avvio della scheda.

Focus: Lettura guidata del codice Python prima dell'esecuzione.

AI Literacy: Comprensione preventiva (Understanding AI)

```
# ============================================================
# FASE 1 - USE
Attivita: "Hello, Micro:bit!"
Obiettivo: primo contatto con lo script, lettura guidata
           PRIMA dell'esecuzione nel simulatore.
============================================================

ISTRUZIONI PER I GRUPPI:
1. NON eseguire ancora il codice nel simulatore MakeCode.
2. Leggete il codice riga per riga insieme al gruppo.
3. Compilate la Griglia di lettura guidata (scheda 02) PRIMA
   di premere play.
4. Solo dopo aver scritto le vostre previsioni, eseguite lo
   script e confrontate il risultato con quanto previsto.
```

Lo studente **analizza** il codice per **anticiparne** il comportamento e **formulare un'ipotesi**.

```
1  def on_forever():
2      basic.show_icon(IconNames.HEART)
3      basic.pause(300)
4      basic.show_icon(IconNames.SMALL_HEART)
5  basic.forever(on_forever)
```

Attività 1 – Primi passi

Riga / istruzione	Cosa pensiamo che faccia	Cosa abbiamo osservato
`def on_forever():`		
`basic.show_icon(IconNames.HEART, 500)`		
`basic.pause(100)`		
`basic.show_icon(IconNames.SMALL_HEART, 500)`		
`basic.pause(100)`		
`basic.forever(on_forever)`		

Domande di riflessione

1. Cosa succederebbe se eliminassimo l'ultima riga '**basic.forever(on_forever)**'?

   _____

2. Cosa cambia visivamente se modifichiamo il valore '**500**'?

   _____

3. Perché le istruzioni hanno un rientro e l'ultima riga no?

   _____

Conferma finale (da compilare dopo l'esecuzione)

Il comportamento osservato corrisponde a quanto previsto dal gruppo?

- ☐ Sì, completamente
- ☐ Parzialmente
- ☐ No

Se la risposta è "parzialmente" o "no", quale parte del codice avevate interpretato in modo diverso? _____

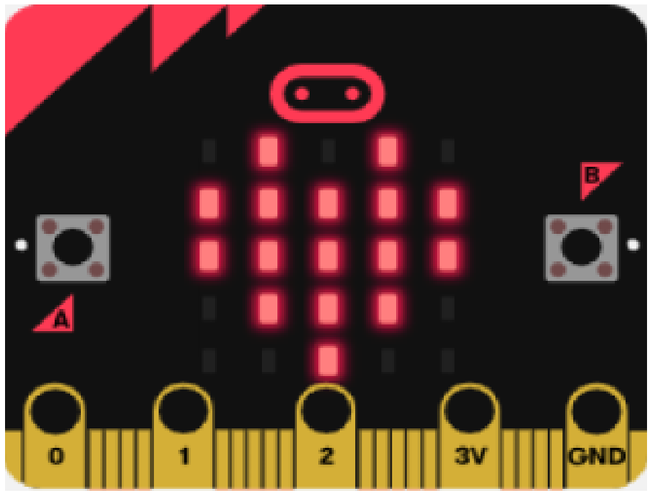
Attività 1 – Primi passi

Use (run and investigate)

Confronta i risultati attesi ed effettivi per **convalidare** o **confutare** l'ipotesi iniziale.

```python
def on_forever():
    pass
    basic.show_leds("""
    # # . # #
    # # # # #
    # # # # #
    . # # # .
    . . # . .
    """)
    basic.pause(100)
    basic.show_leds("""
    . . . . .
    . # . # .
    . # # # .
    . . # . .
    . . . . .
    """)
basic.forever(on_forever)
```

```python
def on_forever():
    basic.show_icon(IconNames.HEART, 300)
    pause(300)
    basic.show_icon(IconNames.SMALL_HEART, 100)
basic.forever(on_forever)
```



Lo studente **esegue** il codice per osservarne il **comportamento effettivo** e testare l'ipotesi.

Attività 1 – Primi passi

Utilizzo di modelli già conosciuti per consolidare

```
1   def on_forever():
2       pass
3       basic.show_icon(IconNames.HEART, 500)
4       basic.pause(100)
5       basic.show_icon(IconNames.SMALL_HEART, 500)
6       basic.pause(100)
7   basic.forever(on_forever)
8
```




```
for i in range(5):
    print("Vai avanti")
```

Indentazione

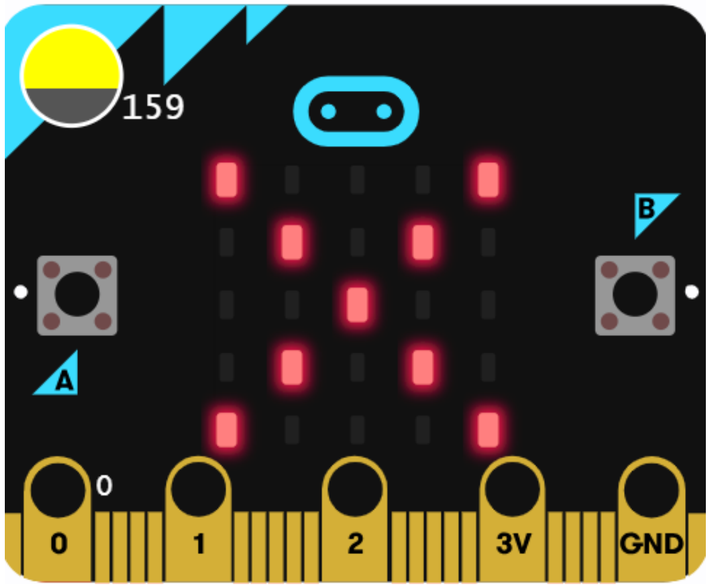


Attività 2 – Sensori

modifica parametri (use and modify)

Attività 2 – Sensori

Modify - analizzare



Attività 2

Rilevatore di buio

Prompt: Lettura luce e accensione LED al buio.

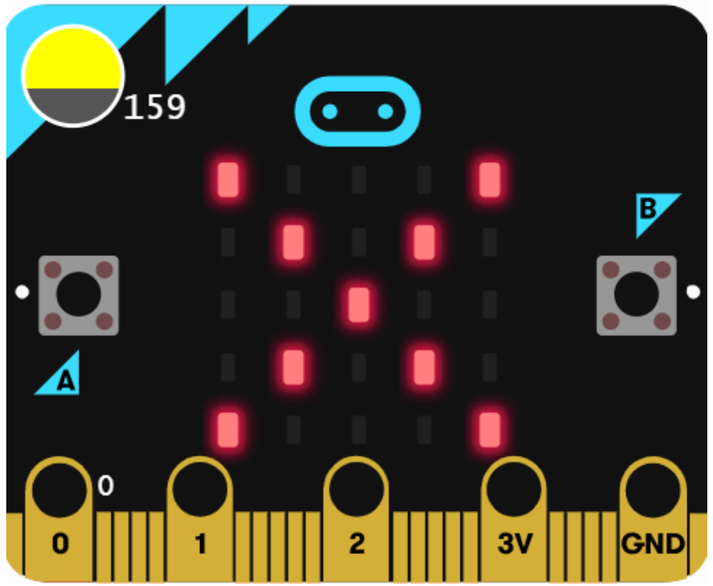
Focus: Modifica della soglia (da 100 a 50) direttamente in Python.

Concetti: Variabili, operatori di confronto e taratura

```
def on_forever():
    if  input.light_level() < 100 :
        basic.show_icon(IconNames.YES)
    else:
        basic.show_icon(IconNames.NO)
    pause(100)
basic.forever(on_forever)
```

Attività 2 – Sensori

Modify - analizzare



Attività 2

Rilevatore di buio

Prompt: Lettura luce e accensione LED al buio.

Focus: Modifica della soglia (da 100 a 50) direttamente in Python.

Concetti: Variabili, operatori di confronto e taratura

```
1   def on_forever():
2       if  input.light_level() < 100 :  #basso livello di luce
3           basic.show_icon(IconNames.YES) #accende il led
4       else:
5           basic.show_icon(IconNames.NO)  #buon livello di luce
6       pause(100)                         #led spento
7   basic.forever(on_forever)
```

Attività 2 – Sensori

Modify - creare

Attività 2

Rilevatore di buio

Prompt: Lettura luce e accensione LED al buio.

Focus: Aggiungi l’accensione del LED esterno

Concetti: Variabili, operatori di confronto e taratura



Tutor micro:bit Python

Incolla il tuo codice MakeCode: ti aiuto a capirlo e a farlo funzionare, senza darti la soluzione.

Nuova chat

led esterno

Perfetto, un LED esterno collegato a un pin della scheda: questa è l'attività del rilevatore di buio con LED esterno.

Per essere sicuro di aver capito bene l'obiettivo: vuoi che il LED esterno si accenda quando c'è poca luce (buio) e si spenga quando c'è abbastanza luce, proprio come fa il display con YES e NO nel tuo codice attuale?

si

Bene, ora che l'obiettivo è chiaro, vediamo cosa fa il tuo codice e cosa ti serve in più per accendere un LED esterno.

Cosa fa già il tuo codice:

- `basic.forever(on_forever)` — categoria **Basic**: ripete la funzione `on_forever` all'infinito, così il controllo della luce avviene in continuazione, non solo una volta.
- `input.light_level()` — categoria **Input**: legge quanta luce c'è nell'ambiente, restituendo un numero da `0` (buio totale) a 255 (molta luce). Tu lo confronti con `100` per decidere se è buio o no.
- `if` / `else` — categoria **Logic**: fa scegliere al programma quale blocco di istruzioni eseguire in base a una condizione (qui, se la luce è sotto 100 oppure no).
- `basic.show_icon(...)` — categoria **Basic**: accende dei LED del display per disegnare un'icona. Qui lo usi per mostrare YES o NO a seconda della luce.

Attività 2 – Sensori

Modify - creare

Attività 2

Rilevatore di buio

Prompt: Lettura luce e accensione LED al buio.

Focus: Aggiungi l’accensione del LED esterno

Concetti: Variabili, operatori di confronto e taratura

Cosa ti serve di nuovo per il LED esterno:

`pins.digital_write_pin(DigitalPin.P0, valore)`

- Categoria: **Advanced > Pins** (si apre cliccando "Advanced" nel menu)
- Cosa fa: accende o spegne un pin della scheda, mandando corrente elettrica (1) o togliendola (0). Se il LED esterno è collegato a quel pin, si accende o si spegne.
- Cosa richiede: il nome del pin a cui è collegato fisicamente il LED (es. `DigitalPin.P0`) e un valore: `1` per accendere, `0` per spegnere.
- Errore tipico: dimenticare di scrivere `0` nel ramo in cui il LED deve spegnersi, lasciandolo acceso anche quando non dovrebbe.

Prova a modificare il tuo codice aggiungendo questi pezzi, poi incollami il risultato e dim
cosa succede nel simulatore.

Attività 2 – Sensori

Modify - creare

Attività 2

Rilevatore di buio

Prompt: Lettura luce e accensione LED al buio.

Focus: Aggiungi l'accensione del LED esterno

Concetti: Variabili, operatori di confronto e taratura

Cosa ti serve di nuovo per il LED esterno:

`pins.digital_write_pin(DigitalPin.P0, valore)`

- Categoria: **Advanced > Pins** (si apre cliccando "Advanced" nel menu)
- Cosa fa: accende o spegne un pin della scheda, mandando corrente elettrica (1) o togliendola (0). Se il LED esterno è collegato a quel pin, si accende o si spegne.
- Cosa richiede: il nome del pin a cui è collegato fisicamente il LED (es. `DigitalPin.P0`) e un valore: `1` per accendere, `0` per spegnere.
- Errore tipico: dimenticare di scrivere `0` nel ramo in cui il LED deve spegnersi, lasciandolo acceso anche quando non dovrebbe.

Prova a modificare il tuo codice aggiungendo questi pezzi, poi incollami il risultato e dim
cosa succede nel simulatore.

```
def on_forever():
    if  input.light_level() < 100 :  #basso livell
        basic.show_icon(IconNames.YES) #accende il l
    else:
        basic.show_icon(IconNames.NO)  #buon livell
    pause(100)                         #led spento
basic.forever(on_forever)
```

```
def on_forever():
    if  input.light_level() < 100 :  #basso livello di luce
        basic.show_icon(IconNames.YES) #accende il led
        pins.digital_write_pin(DigitalPin.P0, 1)
    else:
        basic.show_icon(IconNames.NO)  #buon livello di luce
        pins.digital_write_pin(DigitalPin.P0, 0)
    pause(100)                         #led spento
basic.forever(on_forever)
```

Attività 3 – Contapassi

modifica parametri (Modify)

Attività 3 – Contapassi

Modify - creare

Attività 3

Contatore di passi

Sfida: Segnale ogni 10 passi registrati dall'accelerometro.

Focus: Espansione autonoma di cicli e contatori fornite dall'IA.

Risultato: Ristrutturazione attiva del codice (Modify)

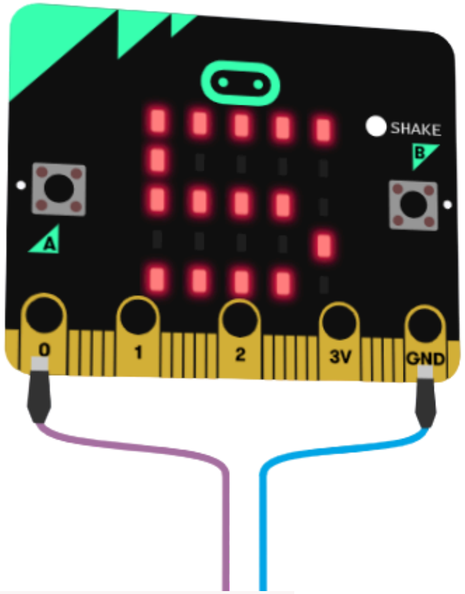
Se dobbiamo contare, dobbiamo creare una variabile, cioè una scatola in cui collezionare tutte le volte in cui accade qualcosa

```
1   scuotimento = 0
2
3   input.on_gesture(Gesture.SHAKE, on_gesture_shake)
4
5   def on_gesture_shake():
6       global scuotimento
7       scuotimento = scuotimento + 1
8       basic.show_number(scuotimento)
```

Legge l'accelerometro

Attività 3 – Contapassi

Modify - creare



Attività 3

Contatore di passi

Sfida: Segnale ogni 10 passi registrati dall'accelerometro.

Focus: Espansione autonoma di cicli e contatori fornite dall'IA.

Risultato: Ristrutturazione attiva del codice (Modify)

```
scuotimento = 0

input.on_gesture(Gesture.SHAKE, on_gesture_shake)

def on_gesture_shake():
    global scuotimento
    scuotimento = scuotimento + 1
    basic.show_number(scuotimento)
```

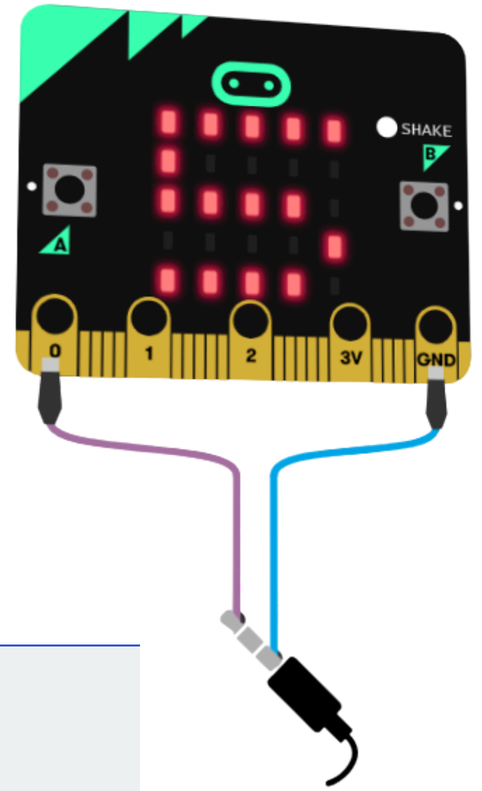
Tutor micro:bit Python • Gem personalizzato

Per suonare ogni 10 passi, ti serve un comando per riprodurre un suono (che trovi nella categoria **Music**, come `music.play_tone`) e un modo per controllare se il numero totale dei passi è un multiplo di 10.

A quale categoria appartiene il comando per calcolare il resto della divisione (il modulo `%`) che usi per scoprire se i passi sono arrivati a 10? E dove pensi vada inserito questo controllo all'interno del tuo codice del contapassi?

Attività 3 – Contapassi

Modify - creare



Attività 3

Contatore di passi

Sfida: Segnale ogni 10 passi registrati dall'accelerometro.

Focus: Espansione autonoma di cicli e contatori fornite dall'IA.

Risultato: Ristrutturazione attiva del codice (Modify)

```
scuotimento = 0

input.on_gesture(Gesture.SHAKE, on_gesture_shake)

def on_gesture_shake():
    global scuotimento
    scuotimento = scuotimento + 1
    basic.show_number(scuotimento)
```

```
scuotimento = 0

def on_gesture_shake():
    global scuotimento
    scuotimento = scuotimento + 1
    basic.show_number(scuotimento)
    if scuotimento % 10 == 0:
        music.tone_playable(523, 1000)

input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

Attività 4 – Semaforo

debugging (Modify)

Attività 4 - Il semaforo

controllo e debugging: le funzioni

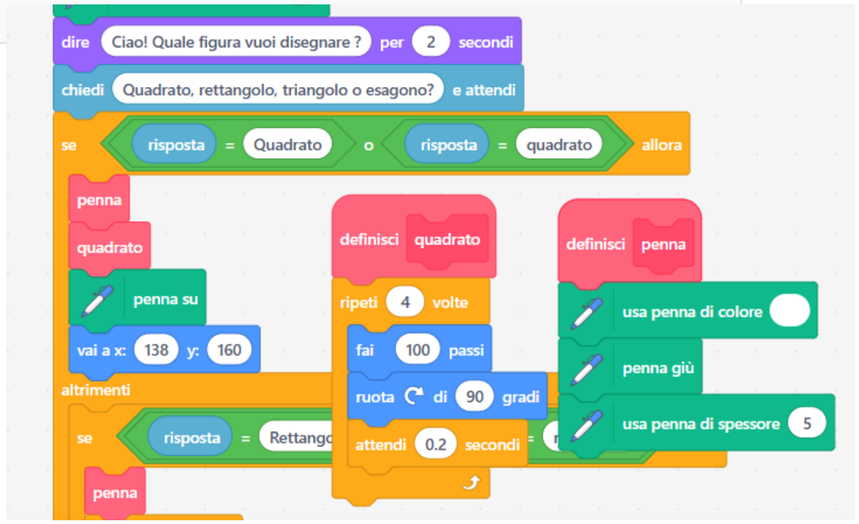
Attività 4

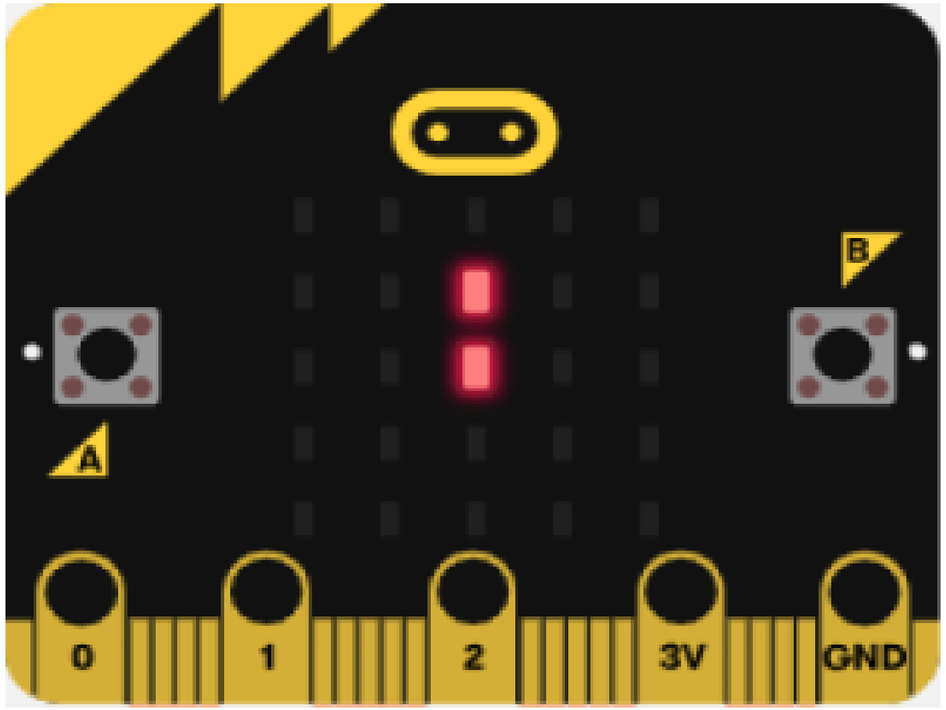
Semaforo interattivo

Scenario: Introduzione di un bug logico intenzionale.

Focus: Debugging guidato e quesiti analitici all'IA.

AI Literacy: Analisi di allucinazioni e limiti dell'IA





Attività 4 - Il semaforo
controllo e debugging

Attività 4

Semaforo interattivo

Scenario: Introduzione di un bug logico intenzionale.

Focus: Debugging guidato e quesiti analitici all'IA.

AI Literacy: Analisi di allucinazioni e limiti dell'IA

```
def mostra_verde():
    # solo verde: accende il LED in basso, spegne gli altri
    led.unplot(2, 0)
    led.unplot(2, 1)
    led.plot(2, 2)

def mostra_verde_arancione():
    # verde e arancione insieme: accende i LED in basso e al centro
    led.unplot(2, 0)
    led.plot(2, 1)
    led.plot(2, 2)

def mostra_rosso():
    # solo rosso: accende il LED in alto
    led.plot(2, 0)
    led.unplot(2, 1)

# Il ciclo principale: si ripete da solo all'infinito

def on_forever():
    mostra_verde()
    basic.pause(3000)
    mostra_verde_arancione()
    basic.pause(2000)
    mostra_rosso()
    basic.pause(3000)
basic.forever(on_forever)
```

Attività 5 e 6 – Il Progetto IoT

Avanzato (Create)

Attività 5 e 6 – Il Progetto IoT

Attività 5: Micro-creazione guidata

Lavoro a gruppi su specifici moduli (sensore umidità, attuatori, radio).

Ruolo IA: Supporto puntuale su specifiche sintassi Python (Applying AI)

Sfida

Nel progetto base, le studentesse e gli studenti, suddivisi in piccoli gruppi, creano in autonomia dei singoli task a scelta tra:

- accensione di un LED collegato al sensore 
- lettura del sensore di umidità 
- invio via radio del dato su un secondo micro:bit 

Ruolo dell'IA

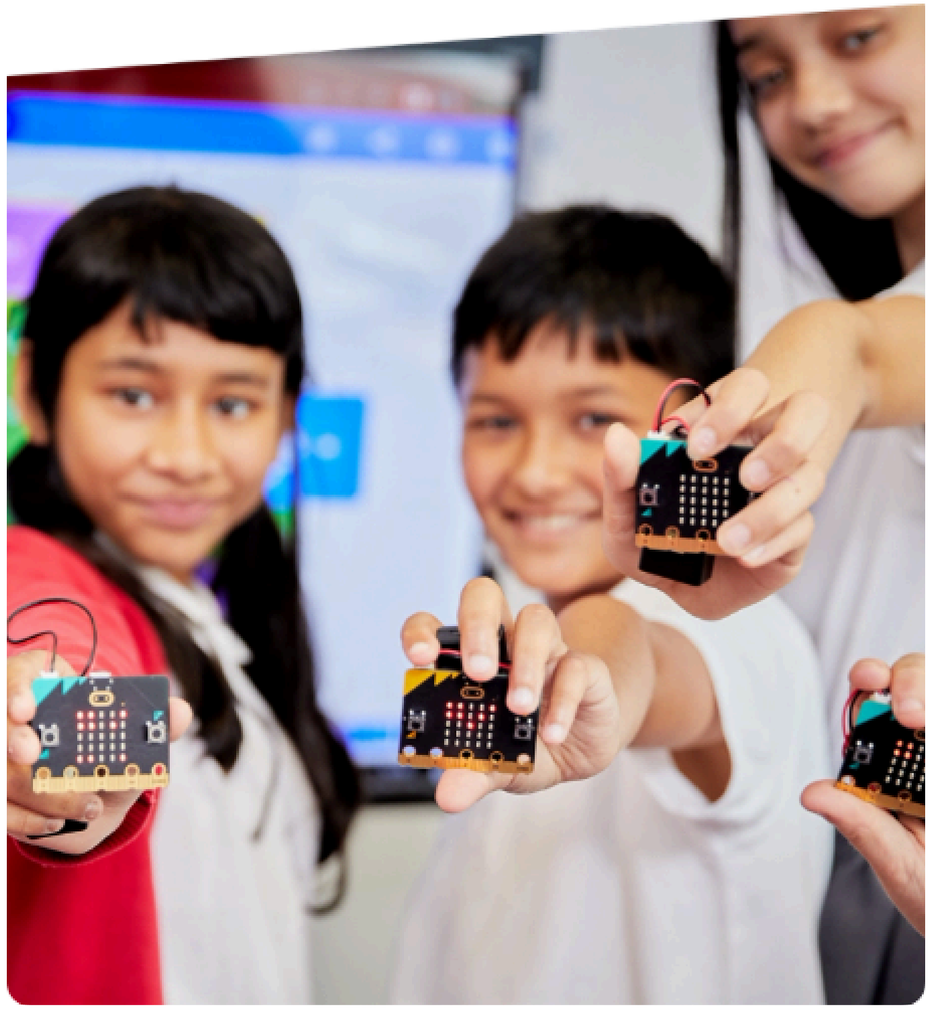
I vari gruppi non chiedono il codice completo, ma, se necessario, formulano prompt specifici (Come invio un numero via radio in Python con micro:bit?), esercitando così anche la dimensione Applying AI.

Attività 5 e 6 – Il Progetto IoT

Attività 6: Sistema Irrigazione Simulato

Sistema integrato con mini-pompa e rete radio tra due schede.

Ruolo IA: Consulente per l'integrazione di sistemi (Creating with AI)



Le tre dimensioni di competenza integrata

DIMENSIONE 1

Tecnico-Algoritmica

Coding Python, hardware micro:bit e debugging strutturato.

DIMENSIONE 2

Critica sull'IA

Prompting mirato, gestione allucinazioni e supervisione umana.

DIMENSIONE 3

Etico-Sociale

Trasparenza, responsabilità e impatto delle decisioni automatizzate.

"L'IA generativa passa da fine tecnologico a mediatore cognitivo: un'esperienza situata al servizio del pensiero critico."

Conclusioni e sviluppi futuri

- **Che cosa emerge dal modello**
  L´IA generativa diventa mediatore cognitivo e oggetto da interrogare, verificare e comprendere: non un modulo teorico separato, ma parte dell'esperienza di programmazione.

- **Perché è utile a scuola**
  Tre dimensioni crescono insieme: competenza tecnico-algoritmica, competenza critica sull´IA, competenza etico-sociale. Il progetto è pensato per essere inclusivo e replicabile.

- **Che cosa manca ancora**
  Sperimentazione sul campo e raccolta di evidenze per verificare le ipotesi sull´apprendimento, il debugging e l'AI Literacy.

Riferimenti essenziali · 1/2

- CINI (2017). Proposta di Indicazioni Nazionali per l’insegnamento dell’Informatica nella Scuola. Roma: CINI.
- Cosgrove, J. & Cachia, R. (2025). DigComp 3.0: European Digital Competence Framework – Fifth Edition.
- D.M. n. 68/2025. Nuove Indicazioni Nazionali per la scuola dell’infanzia e il primo ciclo di istruzione.
- Gerosa, A., Koleszar, V., Tejera, G., Gómez-Sena, L. & Carboni, A. (2022). Educational robotics intervention… Frontiers in Psychology, 13, 904761.
- Lee, I., Martin, F., Denner, J. et al. (2011). Computational thinking for youth in practice. ACM Inroads, 2(1), 32–37.
- Özcan, M. Ş., Çetinkaya, E., Göksun, T. & Kisbu-Sakarya, Y. (2021). Does learning to code influence cognitive skills… British Journal of Educational Psychology, 91(4), 1434–1455.

Riferimenti essenziali · 2/2

- Regolamento (UE) 2024/1689 (AI Act), art. 4 – Alfabetizzazione in materia di IA.
- Robledo-Castro, C., Hederich-Martínez, C. & Castillo-Ossa, L. F. (2023). Cognitive stimulation of executive functions through computational thinking. Journal of Experimental Child Psychology, 235, 105738.
- UNESCO, Miao, F., Shiohira, K. & Lao, N. (2024). AI Competency Framework for Students. Parigi: UNESCO.
- Wang, X., Wan, F. & Dai, J. (2026). Enhancing computational thinking through coding education in primary school students… Frontiers in Psychology, 17, 1734482.
- Raccomandazione del Consiglio dell'Unione Europea C/2024/1030 – Raccomandazione sull'istruzione all'informatica nella scuola.

Grazie per l’attenzione

Giuliana Gelsomino

giuliana.gelsomino23@gmail.com

Barbara Baldi

profbaldi72@gmail.com