



# Crea un chatbot su CodeSkool

## Materiali e strumenti necessari

Software di programmazione a blocchi CodeSkool (<https://ide.codeskool.cc/>). CodeSkool è particolarmente adatto a questo progetto perché combina l'integrazione Open AI con estensioni dei sensori telefonici e funzionalità di sintesi vocale.



Esistono diverse alternative a CodeSkool. Scratch offre estensioni di geolocalizzazione che forniscono funzionalità simili in un ambiente a blocchi. MIT App Inventor include componenti di localizzazione integrati specificamente progettati per lo sviluppo di app mobili. Per le classi con esperienza di programmazione più avanzata, librerie Python come geocoder o geopy offrono servizi di localizzazione sofisticati che possono essere integrati in applicazioni più grandi.

## Codifica e comprensione del programma



### 1 Entra in CodeSkool e memorizza le informazioni GPS utilizzando le variabili

Per implementare la funzionalità GPS, inizia creando un nuovo progetto e accedendo all'editor in CodeSkool.

Prima di accedere ai dati GPS, è necessario definire le posizioni di memoria nel programma in cui verranno memorizzate queste informazioni. Per prima cosa, accedere alla sezione "Variabili" della paletta Blocchi e fare clic su "Crea variabile". Creare una variabile denominata "Latitudine" per memorizzare la posizione nord-sud dell'utente. Questa variabile fungerà da contenitore dedicato che memorizzerà questa informazione specifica in tutto il programma. Quindi, tornare alla sezione "Variabili" e creare una seconda variabile denominata "Longitudine" per memorizzare la posizione est-ovest. Insieme, queste due variabili forniranno una posizione geografica completa che persistrà in memoria.

Ora che i contenitori di memoria per i dati di posizione sono stati creati, accedi alle funzionalità GPS del dispositivo per popolare queste variabili con coordinate reali. Dall'interfaccia principale, accedi al pulsante "Estensioni" situato in fondo alla barra degli strumenti. Nella libreria delle estensioni, cerca "Sensori telefono" e aggiungi questa estensione al progetto per accedere alla funzionalità GPS.

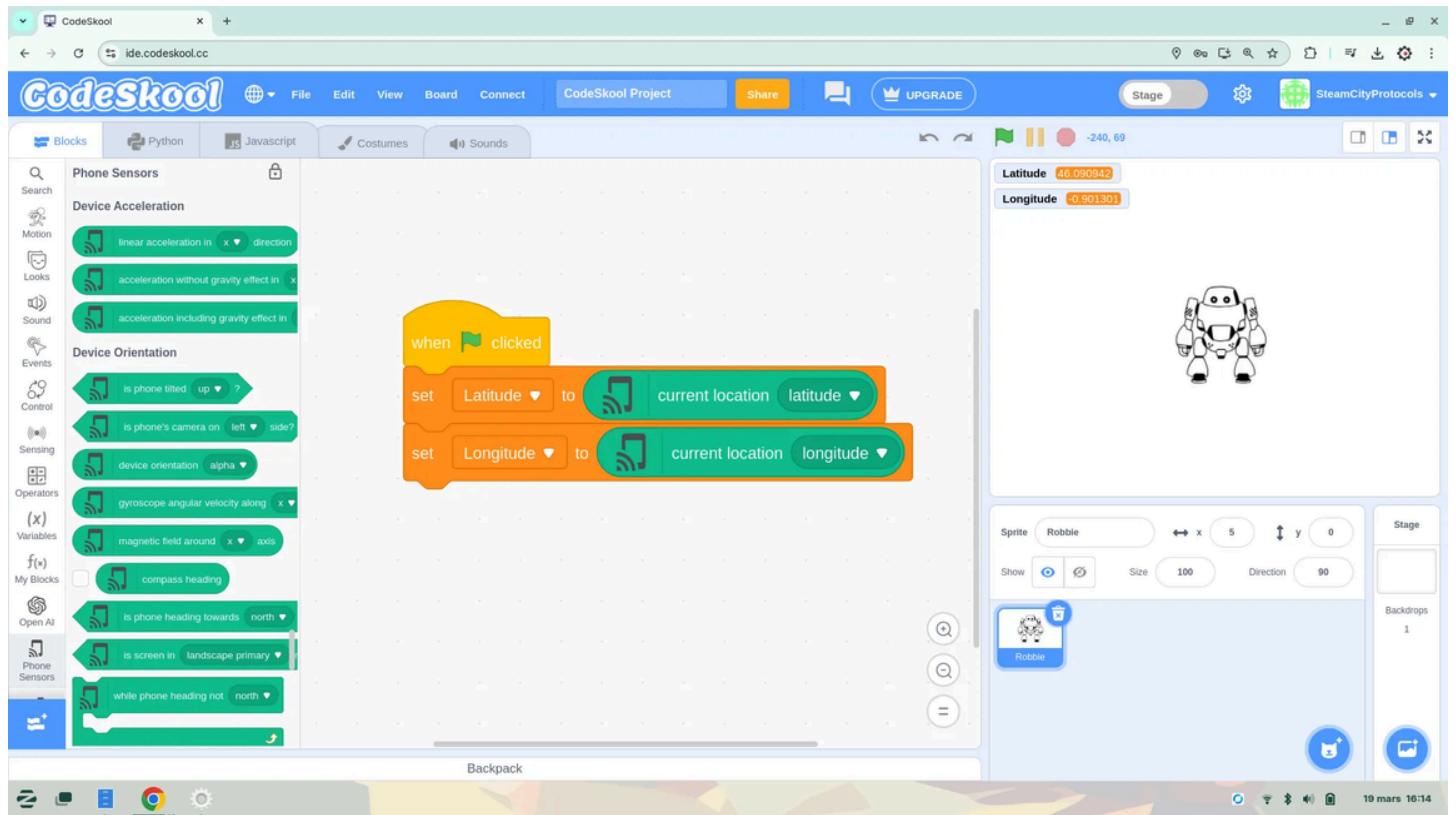


Dopo aver aggiunto l'estensione, trova i blocchi relativi al GPS nella nuova categoria che appare nella tavolozza dei blocchi. Cerca la sottocategoria chiamata "Posizione GPS" in cui troverai il blocco "Posizione attuale" per longitudine e latitudine.

Da ora in poi, attraverso semplici passaggi, vengono identificati tutti i blocchi e le variabili necessarie per creare il programma:

1. Per avviare il processo di raccolta dati, aggiungi un blocco evento "Quando si clicca sul pulsante" all'area di lavoro. Questo servirà come punto di partenza per acquisire la posizione dell'utente.
2. Dalla sezione "Variabili", trascina due blocchi "imposta [variabile] su [valore]" e collegali all'evento del pulsante. Questi blocchi rappresentano il meccanismo attraverso il quale le informazioni passano dai sensori alla memoria del programma.
3. Configurare il primo blocco per definire "Latitudine" e il secondo per definire "Longitudine". La selezione delle variabili dimostra che il programma dispone ora di posizioni di memoria specifiche pronte a contenere diversi tipi di informazioni.

- Collega i blocchi "posizione corrente" appropriati dall'estensione Phone Sensors a ciascun valore di variabile. Seleziona "latitudine" per la prima variabile e "longitudine" per la seconda. Quando questi blocchi vengono eseguiti, acquisiranno le letture dei sensori e le memorizzeranno nella memoria del programma.
- Fare clic sulla bandiera verde nel pannello di test per verificare che le variabili memorizzino correttamente le informazioni.



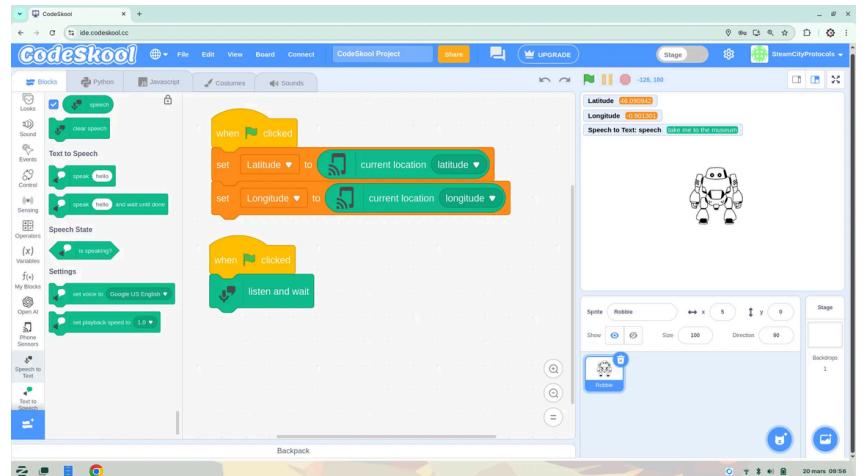
Questo programma funzionerà sia sul computer, sfruttando la posizione della macchina, sia tramite un dispositivo mobile, utilizzando l'estensione Phone Sensors.

## 2

## Implementare e testare il riconoscimento vocale

Sulla base del programma creato, implementare la seconda funzionalità per il riconoscimento vocale.

Nello stesso editor, torna al pulsante "Estensioni" alla fine della barra degli strumenti e cerca "Speech to Text" nella libreria delle estensioni per aggiungerlo al progetto.



Dopo aver aggiunto l'estensione, trova i blocchi relativi al riconoscimento vocale nella nuova categoria che appare nella tavolozza dei blocchi. Cerca il blocco "Ascolta e attendi".

Per maggiore chiarezza nei blocchi, aggiungere un secondo blocco "Quando si clicca sul pulsante" per identificare le due funzionalità di base del sistema.

Una volta implementata la funzionalità di base del riconoscimento vocale, riutilizzare lo stesso metodo di test del passaggio 1 per valutarne sistematicamente l'accuratezza e comprenderne capacità e limiti. Riutilizzare le frasi di test e completare la tabella con le capacità del sistema:

Frase pronunciata	Riconoscimento del sistema	Precisione
<i>La frase formulata dagli studenti</i>	<i>Cosa trascrive il sistema dal tuo discorso</i>	<i>Valutazione della comprensione (Buono / Discreto / Scarso)</i>
<b>"Dov'è il parco più vicino?"</b>		
<b>"Portami al museo"</b>		
<b>"C'è un bar nelle vicinanze?"</b>		
<b>"A che ora vanno al cinema?"</b>		

# 3

## Converti la posizione in città

Implementare la prima integrazione di intelligenza artificiale trasformando le coordinate GPS grezze in un nome di città significativo. Questa funzionalità ha uno scopo importante: converte i dati tecnici (coordinate digitali) in informazioni comprensibili, più intuitive e utili sia per il sistema che per i suoi utenti. Questa conversione servirà anche come base per prompt più complessi nelle attività successive, poiché l'inclusione del nome della città nei prompt futuri aiuterà l'intelligenza artificiale a fornire informazioni più contestualmente rilevanti.

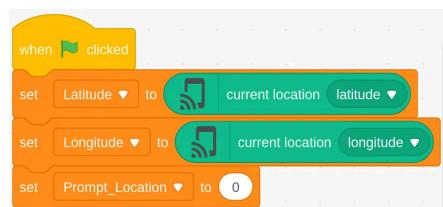
### 1. Preparare le variabili

Per implementare questa conversione da coordinate a città in CodeSkool, prepara il programma utilizzando diverse variabili che possono essere attivate in base alle esigenze. Per prima cosa, crea una nuova variabile chiamata "Location" che memorizzerà il nome della città corrispondente alle coordinate GPS e una seconda variabile chiamata "Prompt\_Location", che conterrà la struttura della domanda da porre al modello di intelligenza artificiale.

### 2. Creare la struttura del prompt

La variabile "Prompt\_Location" dovrebbe consentire di trasformare le variabili di latitudine e longitudine già raccolte dal programma in nomi di città. Per strutturare in modo chiaro questo prompt:

- Nello stesso blocco evento "Quando si fa clic sul pulsante" in cui vengono recuperate le coordinate GPS, aggiungere un blocco "imposta [LocationPrompt] su [valore]" che verrà utilizzato per definire la struttura del prompt.
- Successivamente, definisci la struttura del prompt nel campo [valore]. Inizia formulando la richiesta in modo naturale, offrendo prompt ad alta voce. Potrebbe essere, ad esempio, "Dimmi quale città si trova a questa latitudine e longitudine". Una volta formulate diverse idee, testale nel programma.
- Per combinare testo e variabili in un prompt completo, utilizzare i blocchi "join [valore] [valore]" nella sezione "Operatori". Poiché due variabili (latitudine e longitudine) devono essere integrate con il testo, utilizzare blocchi join annidati:



“

[Dimmi quale città si trova a questa latitudine] + [variabile "Latitudine"] + [e  
longitudine] + [variabile "Longitudine"]

”

Ciò richiede tre blocchi "gruppo [valore] [valore]" annidati insieme. Questa struttura di annidamento è necessaria perché i blocchi join possono combinare solo due elementi alla volta. Annidandoli, si costruisce una frase completa che incorpori più variabili e segmenti di testo:

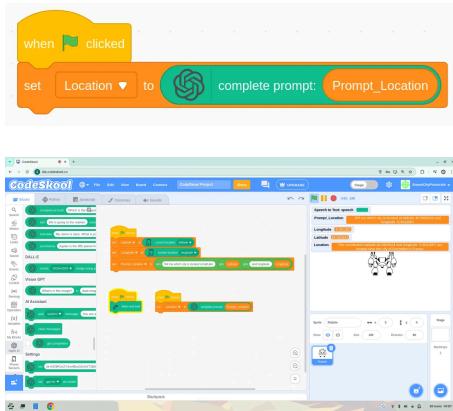
- Primo blocco di unione: Valore 1: Il testo "Dimmi a quale città corrisponde la latitudine" (notare lo spazio alla fine) / Valore 2: Il secondo blocco di unione
- Secondo blocco di join: Valore 1: La variabile "Latitudine" / Valore 2: Il terzo blocco di join
- Terzo blocco di join: Valore 1: Il testo "e longitudine" (notare gli spazi prima e dopo) / Valore 2: La variabile "Longitudine"



### 3. Implementare l'integrazione dell'IA

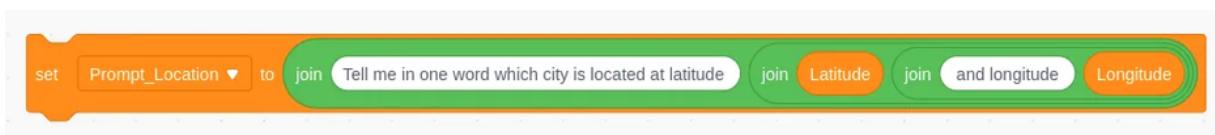
Testa il prompt. Per prima cosa, crea un nuovo blocco "Quando si clicca sul pulsante" per chiarezza e organizzazione nel programma. Questo separa la funzionalità di raccolta dati dall'elaborazione tramite intelligenza artificiale. In questo blocco:

- Aggiungi un blocco "imposta [variabile] su [valore]"
- Imposta la variabile su "Posizione"
- Torna alla scheda "Estensioni" e aggiungi l'estensione "Open AI" al tuo progetto
- Trova il blocco "prompt completo: [prompt]" nell'estensione Open AI
- Imposta questo blocco come valore su "imposta posizione su [valore]"
- Sostituisci il valore predefinito "prompt" con la variabile "Prompt\_Location"

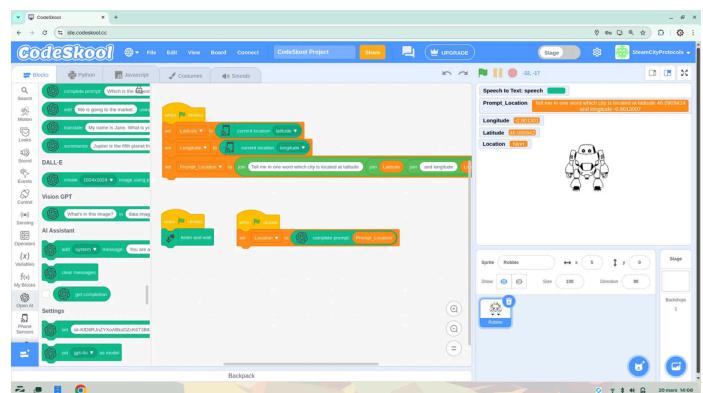


Clicca sulla bandiera verde per testare il programma. Nota che l'IA in genere fornisce risposte prolisse come "La città situata a latitudine 48.8566 e longitudine 2.3522 è Parigi, Francia. Parigi è la capitale e la città più popolosa della Francia". Questa verbosità, sebbene informativa, potrebbe non essere l'ideale per il chatbot che ha bisogno solo del nome della città come componente per i prompt successivi.

Ora perfeziona il prompt per richiedere solo le informazioni specifiche necessarie. Modifica Prompt\_Location per renderlo più specifico: "Dimmi in una parola quale città si trova a latitudine [variabile Latitudine] e longitudine [variabile Longitudine]."



Testando e perfezionando il prompt, imparerai un importante principio dell'ingegneria dei prompt: essere specifici sia riguardo alle informazioni richieste sia al formato di risposta desiderato. Questo processo iterativo dimostra come i prompt possano essere ottimizzati per produrre esattamente le informazioni necessarie, insegnando che un'interazione efficace con l'IA richiede spesso molteplici perfezionamenti per ottenere i risultati desiderati.



# 4

## Crea un chatbot attivato vocalmente

Stabilire la struttura di base dell'assistente vocale implementando una frase di attivazione e un messaggio di saluto.

Per implementarlo in CodeSkool, assicurati che le estensioni "Text to speech" e "Text to speech" siano aggiunte al progetto. Nei blocchi "Text to speech", seleziona il blocco trigger "When I hear [value]" per attivare il chatbot. In questo modo, ogni volta che si clicca sulla bandierina verde, grazie al blocco "Listen and wait" aggiunto all'inizio del protocollo, il chatbot reagirà automaticamente se sente la frase trigger (nell'esempio "Hello Buddy").



Una volta fatto questo, configura le impostazioni vocali per il chatbot. Ottimizzare le impostazioni vocali è essenziale per una comunicazione naturale ed efficace. Le giuste impostazioni per velocità e tipo di voce hanno un impatto significativo sul modo in cui gli utenti comprendono e interagiscono con il chatbot. Pertanto, è necessario utilizzare due blocchi per definire queste impostazioni, disponibili direttamente nella sezione "Text to Speech", nella categoria Impostazioni:

- "Imposta velocità di riproduzione a [valore)": questo blocco controlla la velocità con cui la voce sintetica parla. Il valore varia in genere da 0 (molto lento) a 2 (molto veloce), dove 1 è la velocità normale.
- "scegli voce [valore)": questo blocco consente di selezionare diverse opzioni vocali per l'output di sintesi vocale. Le voci possono variare per genere, età o accento.

•



# 5

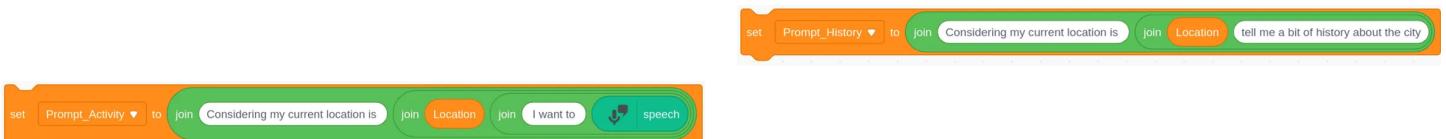
## Progettare flussi di conversazione con prompt

Dopo aver configurato la voce, crea flussi di conversazione strutturati per guidare le interazioni dell'utente con l'assistente. Questa struttura a blocchi logici serve da esempio. Sperimenta diversi approcci organizzativi, imparando dal modo in cui l'assistente risponde per creare codice e prompt migliori.

### Esempio: Ciao amico!

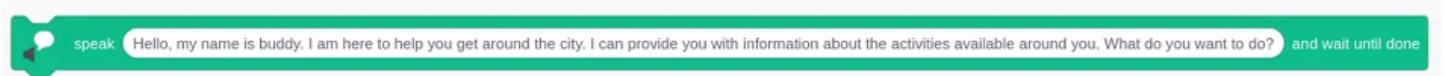
Creare variabili per memorizzare diversi prompt di intelligenza artificiale per argomenti specifici, come fatto in precedenza per la localizzazione. Il nostro agente deve:

- Suggerisci attività nelle vicinanze: abbiamo creato la variabile "Prompt\_Activity" utilizzando 3 blocchi di unione come: "Considerando che la mia posizione attuale è [variabile di posizione], voglio [discorso]"
- Fornisci informazioni storiche sulla città attuale: abbiamo creato la variabile "Prompt\_History" utilizzando 2 blocchi di join come: "Considerando che la mia posizione attuale è [variabile di posizione], raccontami un po' di storia della città"

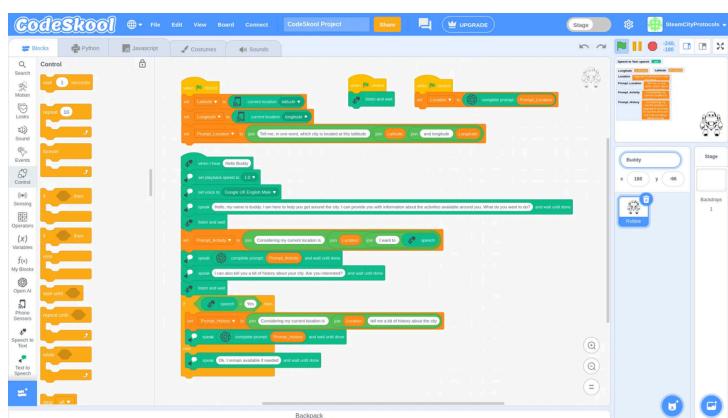


Implementare un semplice albero di dialogo che risponda all'input vocale:

- Una volta che l'utente pronuncia la frase di attivazione "Ciao Buddy", Buddy risponde con un primo suggerimento sulle attività nelle vicinanze utilizzando il seguente messaggio (con un blocco di sintesi vocale): "Ciao, mi chiamo Buddy. Sono qui per aiutarti a muoverti in città. Posso fornirti informazioni sulle attività disponibili nei dintorni. Cosa ti piacerebbe fare?"



- Dopo che Buddy ha finito di parlare, utilizziamo un blocco "ascolta e attendi" che consente all'utente di specificare l'attività desiderata tramite il blocco "imposta Prompt\_Activity".
- In base alla risposta dell'utente, ad esempio "mangia una pizza", Buddy utilizza una combinazione di blocchi "parla" e "completa richiesta" di Open AI per fornire una risposta dettagliata alla richiesta.
- Al termine della discussione, Buddy passa a fornire informazioni sulla storia della città. L'interazione continua con Buddy che chiede: "Posso anche raccontarti un po' della storia della tua città. Ti interessa?"
- A differenza dell'interazione con l'attività, che utilizzava una domanda aperta, qui implementiamo una domanda chiusa che richiede un blocco di controllo "if [ ] then". L'utente può rispondere solo sì o no, attivando risposte diverse dal bot.
  - Se l'utente risponde affermativamente, Buddy presenta la storia della città.
  - Se l'utente risponde di no, Buddy conclude educatamente la conversazione: "Okay. Sono disponibile se necessario."



## Esempio

Ecco un'altra versione del programma basata sugli stessi blocchi, che aggiunge funzioni e capacità di storytelling utilizzando più oggetti di scena e variabili. Per aprirla, vai su ide.codeskool.cc e seleziona "Apri progetto dal link".

Puoi usare il link di condivisione: <https://share.codeskool.cc/eT4U>

Ecco il nostro esempio completo che si basa su tutte le attività di programmazione svolte nei passaggi precedenti disponibile a questo link: <https://share.codeskool.cc/cqZu>

Ecco un video dimostrativo (in francese): [https://drive.google.com/file/d/19cR5d5\\_sKCEtvqY\\_\\_fR5\\_TJbMsap-Knd/view?usp=drivesdk](https://drive.google.com/file/d/19cR5d5_sKCEtvqY__fR5_TJbMsap-Knd/view?usp=drivesdk)

