

Paolo Dalprato

ChoiceMap

Ultimo aggiornamento: 26 gennaio 2026

Un framework per documentare processi a scelte multiple.

ChoiceMap trasforma workflow complessi in esperienze interattive. Invece di leggere un documento dall'inizio alla fine, l'utente naviga attraverso un albero decisionale dove ogni scelta apre un percorso diverso.

L'utente vede una serie di schede con scelte multiple che portano a percorsi diversi. In qualsiasi momento può aprire la mappa visuale per vedere la struttura completa, i nodi già visitati e la posizione attuale. Questa doppia modalità, navigazione guidata e visione d'insieme, riduce il disorientamento tipico dei contenuti ramificati.

The screenshot shows the 'Welcome to the Journey' screen of the ChoiceMap application. At the top left is the text 'Choose Your Path'. At the top right is a circular icon with a tree and the text 'MySelf'. Below the title, there is a message: 'This is an **interactive exploration** where your choices shape the path ahead.' followed by 'There are no right or wrong answers, only different perspectives to discover.' A callout box contains the text 'Each choice opens new possibilities. You can always return to explore other paths.' Below this, there is a section titled 'Choose your starting direction:' with three blue buttons labeled 'Path of Knowledge', 'Path of Action', and 'Path of Reflection'. Underneath these buttons is a section titled 'RESOURCES' with a button labeled 'About this project'. At the bottom left is a progress indicator 'Step 1 / 11'. At the bottom right is a button labeled 'View Map' with a map icon.

(<https://docs.ai-know.pro/choicemap/img/navigator-interfaccia.png>)

Lo strumento è pensato per formatori aziendali, instructional designer e creatori di contenuti didattici che vogliono trasformare materiali statici in esperienze coinvolgenti. Un manuale di procedure diventa un percorso guidato, un troubleshooting diventa una guida interattiva, un caso di studio diventa una simulazione decisionale.

Su GitHub il repository è github.com/paolodalprato/ChoiceMap (<https://github.com/paolodalprato/ChoiceMap>)

Cosa puoi creare

Con ChoiceMap puoi costruire diversi tipi di contenuti interattivi.

Percorsi formativi che si adattano alle risposte dell'utente: chi risponde correttamente prosegue, chi sbaglia riceve spiegazioni aggiuntive prima di continuare.

Simulazioni di processo che guidano attraverso procedure complesse, come l'escalation di un ticket di assistenza o la gestione di una richiesta cliente.

Quiz ramificati dove ogni risposta porta a un feedback specifico e il percorso cambia in base alle scelte.

Storie interattive in stile "libro-game" per onboarding, formazione sulla sicurezza, o esplorazione di scenari aziendali.

Come funziona

Il framework si compone di tre strumenti che lavorano insieme:

- **Navigator:** visualizza lo scenario e gestisce l'interazione per l'utente.
- **Scenario Editor:** permette di creare e modificare i contenuti in modo visuale.
- **Theme Editor:** consente di personalizzare colori, font e branding aziendale.

→ [Prova il Navigator](https://paolodalprato.github.io/ChoiceMap/choicemap.html) (<https://paolodalprato.github.io/ChoiceMap/choicemap.html>)

I contenuti sono file JSON semplici, modificabili anche con un editor di testo. Non serve un server complesso: basta un web server statico per pubblicare gli scenari, ad esempio GitHub Pages.

A chi è rivolto questo manuale

Il manuale guida passo dopo passo dalla prima installazione fino alla pubblicazione online. Non richiede competenze di programmazione. È sufficiente saper usare un browser e avere dimestichezza con operazioni base come scaricare file e modificare testo.

Se sai già cos'è un file JSON, troverai la struttura dei dati familiare. Se non lo sai, il manuale spiega tutto quello che serve per creare scenari funzionanti.

Indice

Guida completa

- Cosa puoi creare
- Come funziona
- A chi è rivolto questo manuale

Panoramica

- I tre componenti
- Casi d'uso tipici
- Architettura tecnica

Installazione e avvio

- Requisiti
- Scaricare il pacchetto
- Avvio su Windows
- Avvio manuale
- Perché serve un server
- Porte e conflitti
- Struttura delle cartelle

Interfaccia del Navigator

- Area intestazione
- Indicatore di progresso
- Mappa
- Area contenuto
- Pulsanti di scelta
- Navigazione
- Mappa del percorso

- Responsività

Interfaccia dello Scenario Editor

- Come si presenta
- I blocchi dell'area di editing
- Settings: metadati e traduzioni
- Il file JSON
- Salvataggio

Interfaccia del Theme Editor

- Layout generale
- Colonna di sinistra
- Anteprima live
- Flusso di lavoro tipico

Creare uno scenario

- Pianificare prima di costruire
- Workflow consigliato
- Gestire i nodi terminali
- Gestire i nodi orfani
- Eliminare nodi in sicurezza
- Creare loop
- Aggiungere risorse
- Modificare uno scenario esistente
- Dalla creazione alla pubblicazione

Struttura dei contenuti

- Architettura dei file
- Il file scenario
- Validazione

Personalizzare il tema

- Workflow consigliato
- Consigli per il branding
- Consigli per la tipografia
- Consigli per i colori
- Salvare e applicare
- Gestire temi multipli
- Modificare il tema via JSON

Pubblicazione

- Ruoli e responsabilità
- Workflow dell'amministratore
- Preparare la pubblicazione
- Destinazioni di pubblicazione
- GitHub Pages
- Gestire scenari multipli
- Embedding in altri siti
- Sicurezza e accesso

Riferimento rapido

- Avviare il server locale
- Creare uno scenario
- Modificare uno scenario esistente
- Personalizzare il tema
- Configurare config.json
- Testare in locale
- Pubblicare
- Codice colori mappa (Scenario Editor)
- Markdown nel contenuto
- Troubleshooting veloce

Panoramica

ChoiceMap trasforma contenuti lineari in esperienze interattive. Invece di leggere un documento dall'inizio alla fine, l'utente naviga attraverso un albero di scelte dove ogni decisione apre un percorso diverso.

I tre componenti

Il framework è composto da tre strumenti distinti, ognuno con un ruolo specifico.

Navigator

È il motore di visualizzazione, quello che l'utente finale vede e usa. Carica uno scenario JSON e lo presenta come una sequenza di schermate. Ogni schermata mostra un contenuto testuale e una serie di pulsanti tra cui scegliere, con la possibilità di allegare file e condividere link. La scelta determina quale schermata appare dopo.

The screenshot shows the 'Navigator' interface of ChoiceMap. At the top left is the text 'Choose Your Path'. At the top right is a logo with a stylized tree and the text 'MySelf'. The main heading 'Welcome to the Journey' is centered. Below it, a message reads: 'This is an **interactive exploration** where your choices shape the path ahead.' Another message states: 'There are no right or wrong answers, only different perspectives to discover.' A callout box contains the text: 'Each choice opens new possibilities. You can always return to explore other paths.' Below this, a section titled 'Choose your starting direction:' contains three blue buttons labeled 'Path of Knowledge', 'Path of Action', and 'Path of Reflection'. Underneath these buttons is a section titled 'RESOURCES' with a single link: 'About this project'. At the bottom left is the text 'Step 1 / 11'. At the bottom right is a button labeled 'View Map' with a map icon.

(<https://docs.ai-know.pro/choicemap/img/navigator-interfaccia.png>)

Il Navigator include una mappa visuale del percorso che mostra dove si trova l'utente nell'albero decisionale, quali nodi ha già visitato e quali restano da esplorare. L'utente può tornare indietro, ricominciare da capo, o saltare a un punto specifico della parte già visitata della mappa.



(<https://docs.ai-know.pro/choicemap/img/navigator-mappa.png>)

Scenario Editor

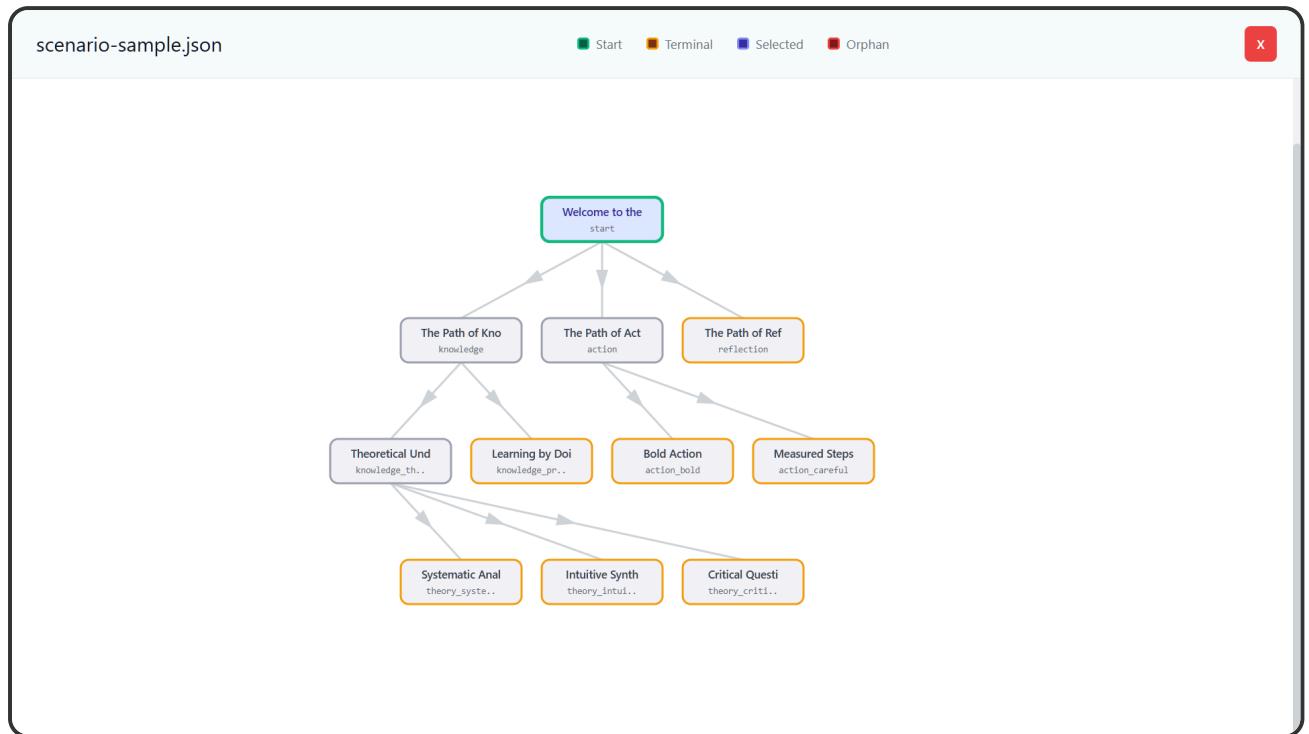
È lo strumento per creare e modificare scenari senza scrivere JSON a mano. Se si apre per modificarlo uno scenario esistente, l'editor presenta una lista di tutti i nodi, permette di modificarne il contenuto, aggiungere scelte, collegare nodi tra loro.

The Scenario Editor interface shows a JSON file named "scenario-sample.json" containing 11 nodes. On the left, a sidebar lists nodes with their names, descriptions, and status (START, END). The main area displays the "Content" and "Choices" sections for the "Node start" node.

- Content:**
 - Content - Markdown
 - # Welcome to the Journey
 - This is an **interactive exploration** where your choices shape the path ahead.
 - There are no right or wrong answers, only different perspectives to discover.
- Choices (3):**
 - Choice 1:** Path of Knowledge → Target Node: knowledge
 - Choice 2:** Path of Action → Target Node: action
 - Choice 3:** Path of Reflection → Target Node: reflection

(https://docs.ai-know.pro/choicemap/img/scenario_edit.png)

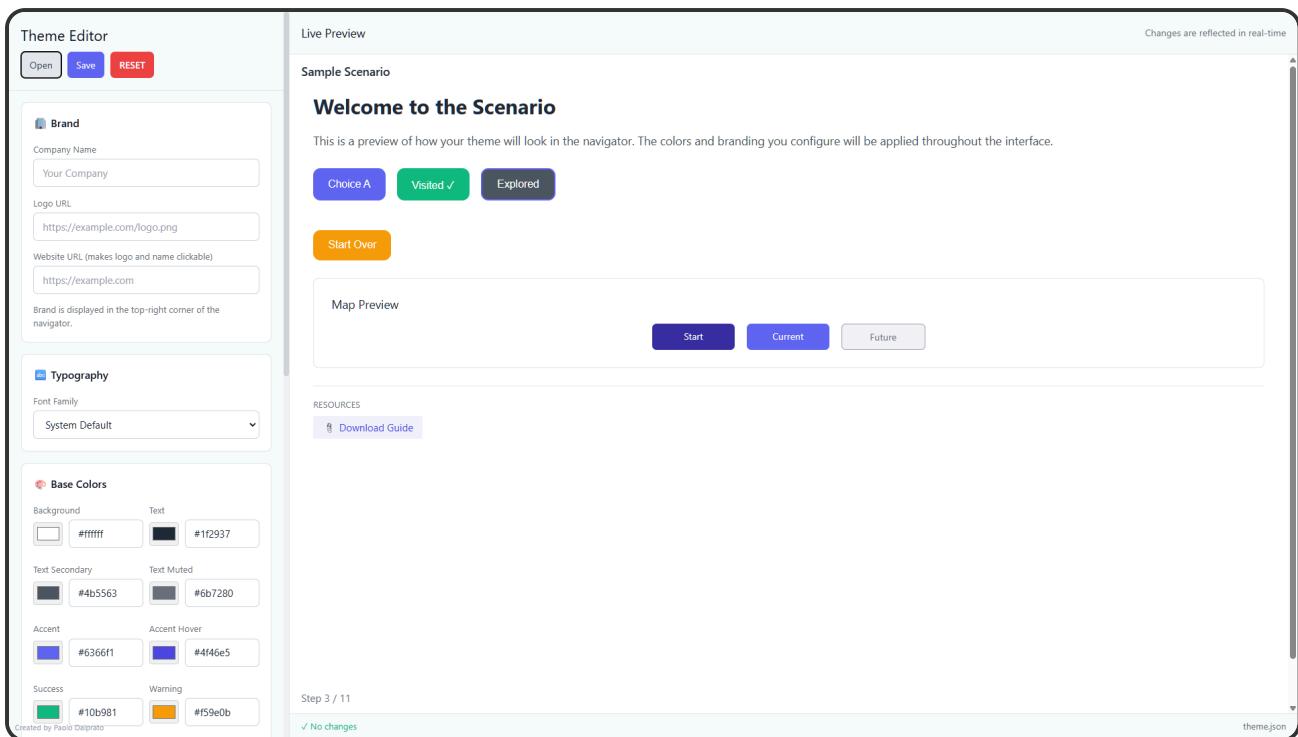
L'editor include una vista mappa che mostra la struttura ad albero dello scenario, evidenziando il nodo iniziale, i nodi finali e eventuali nodi "orfani" (scollegati dal flusso principale). Questa vista aiuta a individuare problemi strutturali prima di pubblicare.



(https://docs.ai-know.pro/choicemap/img/scenario_mappa.png)

Theme Editor

È lo strumento per personalizzare l'aspetto visivo del Navigator. Permette di modificare colori, font, aggiungere logo e nome dell'azienda, senza toccare il codice.



(https://docs.ai-know.pro/choicemap/img/tema_editor.png)

Le modifiche si vedono in tempo reale in un'anteprima. Una volta soddisfatti, si salva il tema come file JSON e lo si associa allo scenario.

Casi d'uso tipici

ChoiceMap si adatta a diversi contesti formativi e comunicativi.

Formazione aziendale

Un corso di onboarding può diventare un percorso interattivo dove il nuovo dipendente esplora le procedure aziendali facendo scelte e ricevendo feedback contestuali. Invece di leggere un manuale di 50 pagine, naviga attraverso scenari che simulano situazioni reali, connettendo ogni passaggio ai capitoli specifici nei documenti correlati.

La formazione sulla compliance si presta particolarmente bene: scenari che presentano situazioni ambigue, chiedono una decisione, e mostrano le conseguenze di scelte corrette o errate.

Quiz e valutazioni

Un quiz tradizionale presenta domande in sequenza. Con ChoiceMap il percorso cambia in base alle risposte: chi risponde correttamente procede, chi sbaglia riceve una spiegazione e può riprovare o viene indirizzato a un percorso di approfondimento.

Questo approccio trasforma la valutazione in un'esperienza di apprendimento: l'errore non è solo un punteggio negativo ma un'opportunità per capire meglio.

Documentazione di processo

Un workflow come la gestione di un ticket di assistenza può essere rappresentato come un albero decisionale. L'operatore risponde a domande sulla situazione ("È un problema tecnico o di fatturazione?", "Il servizio è completamente fermo?") e viene guidato verso la procedura corretta.

Questo approccio permette di rappresentare situazioni complesse, riducendo gli errori e accelerando la formazione di nuovo personale.

Storytelling interattivo

Storie in stile "libro-game" dove il lettore fa scelte fra percorsi multipli. Utile per formazione esperienziale, simulazioni di negoziazione, esplorazione di scenari etici.

Architettura tecnica

ChoiceMap è costruito con tecnologie web standard: HTML, CSS, JavaScript. Usa React per l'interfaccia e Tailwind CSS per lo stile, entrambi caricati da CDN. Non richiede compilazione né installazione di dipendenze.

Ogni scenario è un file JSON con una struttura semplice: metadati (titolo, autore), traduzioni delle etichette dell'interfaccia, e una collezione di nodi. Ogni nodo ha un contenuto testuale in Markdown e una lista di scelte che puntano ad altri nodi.

I temi sono anch'essi file JSON che definiscono colori, font e informazioni di branding.

Il Navigator funziona su qualsiasi browser moderno e si adatta automaticamente a desktop e dispositivi mobili.

Installazione e avvio

Questa sezione dettaglia i requisiti e le procedure per far funzionare ChoiceMap sul proprio computer.

Requisiti

ChoiceMap ha requisiti minimi. Serve un browser moderno (Chrome, Firefox, Safari, Edge nelle versioni recenti) e Python per avviare il server locale.

Python è necessario solo per lo sviluppo locale. Una volta pubblicato su un server web, gli utenti finali non hanno bisogno di Python: accedono direttamente dal browser.

Su Windows, Python potrebbe non essere preinstallato. Si può scaricare da [python.org](https://www.python.org) o installare tramite Microsoft Store. Per verificare se è presente, aprire il Prompt dei comandi e digitare `python --version`. Se compare un numero di versione (3.x), Python è installato.

Su Mac, Python è tipicamente già presente. Su Linux, è quasi sempre disponibile di default.

Scaricare il pacchetto

Il pacchetto completo è disponibile come file ZIP dalla pagina GitHub del progetto, nella sezione Releases. Il download include:

- `choicemap.html` - il Navigator
- `scenario-editor.html` - l'editor per creare scenari
- `theme-editor.html` - l'editor per personalizzare i temi
- `config.json` - file di configurazione
- `defaults.json` - valori predefiniti condivisi
- `theme.json` - tema predefinito
- `scenario-sample.json` - scenario di esempio narrativo
- `scenario-quiz.json` - esempio di quiz
- `scenario-workflow.json` - esempio di workflow
- Script batch per Windows
- Cartella `docs/` per risorse scaricabili

Estrarre lo ZIP in una cartella a scelta. Il percorso non deve contenere caratteri speciali o spazi per evitare problemi con alcuni sistemi.

Avvio su Windows

Il modo più semplice su Windows è usare gli script batch inclusi.

Fare doppio clic su `start-navigator.bat` per avviare il server e aprire il Navigator nel browser predefinito.
Lo script avvia un server sulla porta 8000.

Fare doppio clic su `start-scenario-editor.bat` per aprire direttamente lo Scenario Editor.

Fare doppio clic su `start-theme-editor.bat` per aprire il Theme Editor.

Se compare un errore che indica che Python non è trovato, è necessario installarlo o aggiungerlo al PATH di sistema.

Gli script creano un server che resta attivo finché la finestra del terminale è aperta. Chiudendo la finestra si ferma il server.

Avvio manuale

Se si preferisce non usare gli script o si è su Mac/Linux, l'avvio manuale richiede pochi comandi.

Aprire un terminale nella cartella del progetto. Su Windows si può fare clic destro tenendo premuto Shift e selezionare "Apri finestra PowerShell qui" o "Apri nel terminale".

Eseguire il comando:

```
python -m http.server 8000
```

Il terminale mostra un messaggio come "Serving HTTP on 0.0.0.0 port 8000". Il server è attivo.

Aprire il browser e navigare agli indirizzi:

- Navigator: <http://localhost:8000/choicemap.html>
- Scenario Editor: <http://localhost:8000/scenario-editor.html>
- Theme Editor: <http://localhost:8000/theme-editor.html>

Per fermare il server, tornare al terminale e premere Ctrl+C.

Perché serve un server

Aprendo i file HTML direttamente (doppio clic o trascinando nel browser), il Navigator non funziona. Appare una schermata vuota o un errore.

Il motivo è una restrizione di sicurezza dei browser. Quando un file HTML viene aperto dal filesystem locale (URL che inizia con `file://`), il browser impedisce di caricare altri file come i JSON degli scenari. È una protezione contro script malevoli.

Il server locale aggira questa limitazione servendo i file tramite HTTP (URL che inizia con `http://`). È lo stesso protocollo usato dai siti web, quindi il browser permette il caricamento.

Questa limitazione esiste solo in locale. Una volta pubblicato su un server web (GitHub Pages, hosting aziendale), tutto funziona normalmente.

Porte e conflitti

Il server usa di default la porta 8000. Se quella porta è già occupata da un altro programma, appare un errore.

Per usare una porta diversa:

```
python -m http.server 8080
```

Ricordare di aggiornare l'indirizzo nel browser: `http://localhost:8080/choicemap.html`.

Struttura delle cartelle

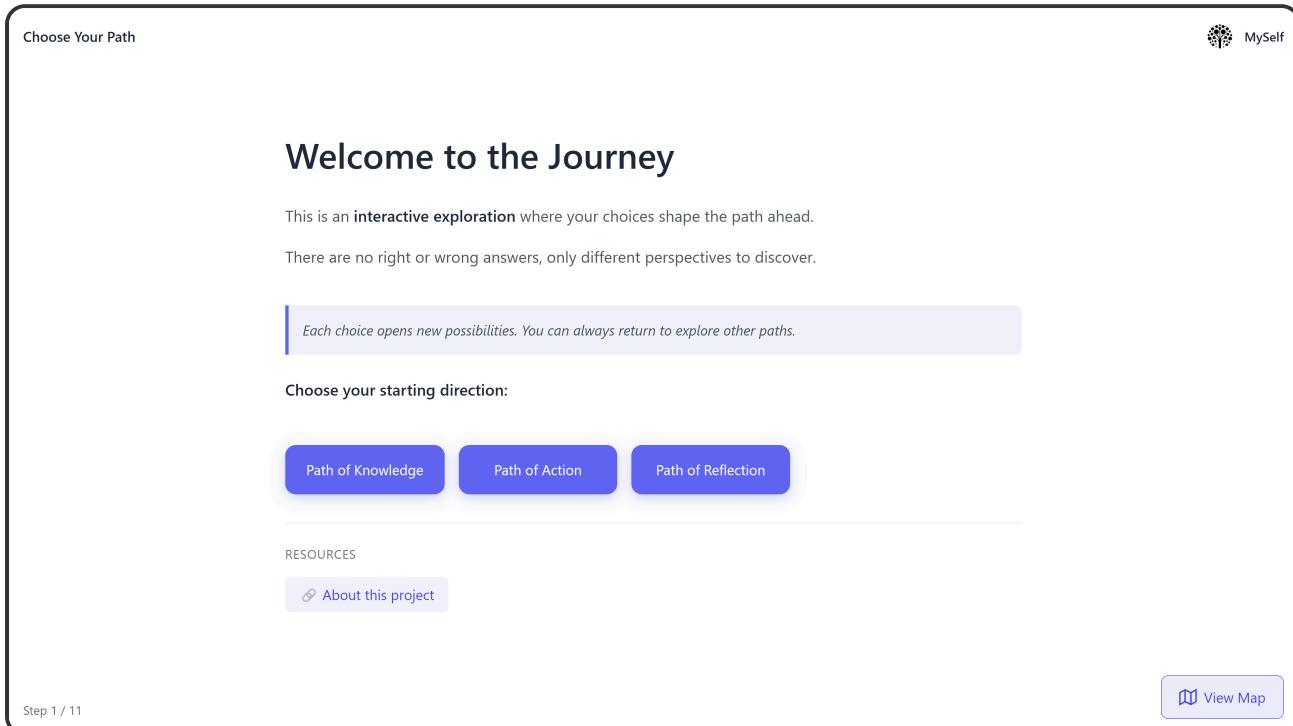
Non è necessario mantenere tutti i file nella stessa cartella, ma semplifica la gestione. La struttura consigliata è:

```
choicemap/
└── choicemap.html
└── scenario-editor.html
└── theme-editor.html
└── config.json
└── defaults.json
└── theme.json
└── scenario-mio-progetto.json
└── theme-azienda.json
└── docs/
    └── (file PDF, risorse)
```

Se si preferisce organizzare diversamente, ad esempio mettendo i file JSON in una sottocartella, è necessario aggiornare i percorsi in config.json.

Interfaccia del Navigator

Il Navigator è ciò che l'utente finale vede quando interagisce con uno scenario. L'interfaccia è progettata per essere pulita e focalizzata sul contenuto, senza distrazioni.



(<https://docs.ai-know.pro/choicemap/img/navigator-interfaccia.png>)

Area intestazione

Nella parte superiore della schermata si trova l'intestazione con le informazioni di contesto.

A sinistra appare il titolo dello scenario, quello definito nei metadati del file JSON. Serve a identificare in quale percorso ci si trova, utile quando un'organizzazione ha più scenari da gestire.

A destra, se configurato, appare il branding aziendale: logo, nome dell'azienda, eventualmente un link al sito.

Indicatore di progresso

In basso a sinistra c'è l'indicatore di progresso, una riga che mostra "Passo X" dove X è il numero di schermate traversate dall'inizio. Questo dà all'utente un senso di avanzamento nel percorso.

L'etichetta "Passo" è personalizzabile nelle traduzioni dello scenario. Per un quiz potrebbe essere "Domanda", per un workflow potrebbe essere "Fase".

Mappa

In basso a destra c'è il pulsante per attivare la visualizzazione della mappa, trattata in modo approfondito più avanti in questo capitolo.

Area contenuto

La sezione centrale è dedicata al contenuto del nodo corrente. Il testo viene renderizzato come Markdown, quindi supporta titoli, grassetto, corsivo, elenchi puntati e citazioni.

Il contenuto può iniziare con un titolo grande (H1) che fa da intestazione della schermata, seguito da paragrafi di testo, elenchi, citazioni evidenziate. La formattazione aiuta a strutturare informazioni complesse in modo leggibile.

Sotto il contenuto testuale possono apparire risorse aggiuntive: link a documenti, file da scaricare, video da guardare. Ogni risorsa ha un'icona che ne indica il tipo e un pulsante per accedervi.

Pulsanti di scelta

Nella parte inferiore dell'area contenuto appaiono i pulsanti di scelta. Ogni pulsante rappresenta un'opzione che l'utente può selezionare, con un testo descrittivo che anticipa dove porterà quella scelta.

I pulsanti cambiano colore in base allo stato. Il colore primario indica scelte non ancora esplorate. Un colore diverso indica scelte già fatte in precedenza, utile quando l'utente torna indietro per esplorare percorsi alternativi. Un terzo colore può indicare scelte esplorate ma non nel percorso corrente.

Quando un nodo non ha scelte, significa che è un punto finale del percorso. Al posto dei pulsanti di scelta appare un messaggio come "**Fine del percorso**" (anche questo testo è modificabile nell'editor degli scenari) e un pulsante per ricominciare.

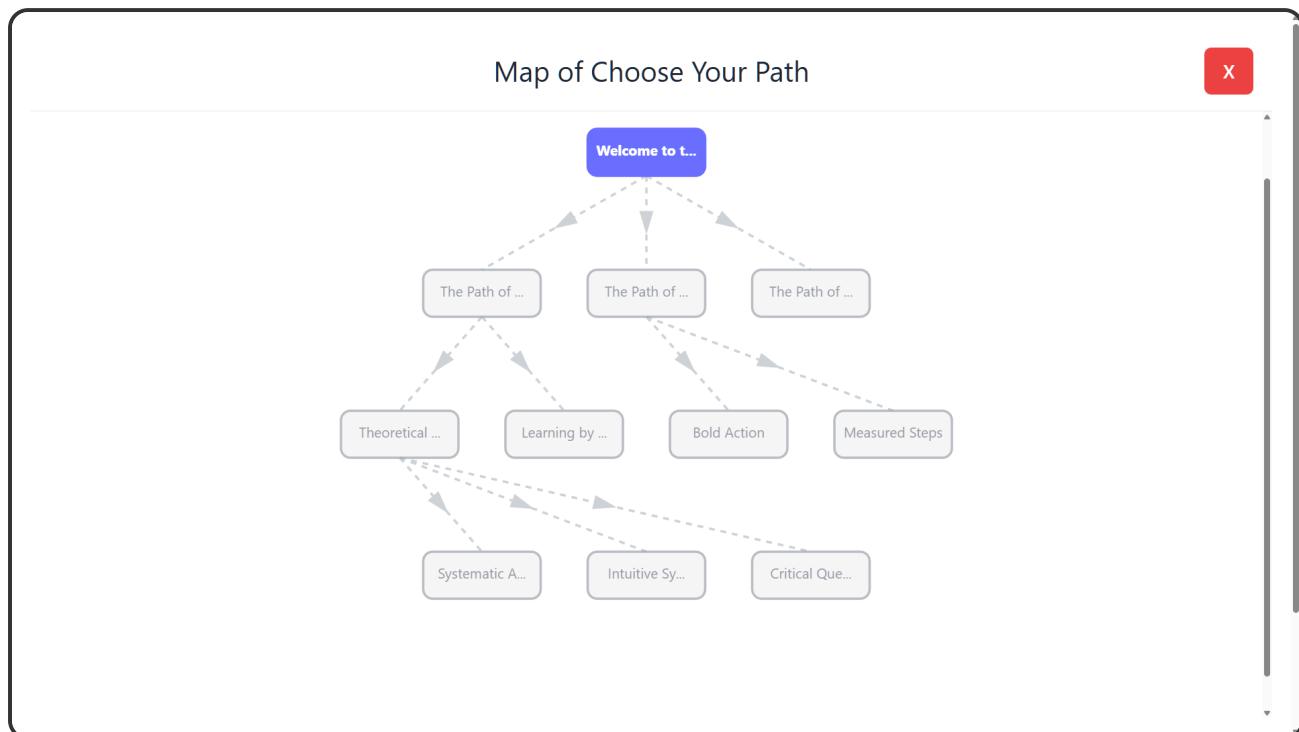
Navigazione

Quando si è iniziato a percorrere uno scenario in alto compare il tasto "**Indietro**", permette di tornare alla scheda precedente, ripercorrendo il percorso fatto. È utile per rileggere informazioni o riconsiderare una scelta.

Il pulsante "**Ricomincia**" riporta all'inizio dello scenario, azzerando il percorso. Utile per esplorare da capo con scelte diverse.

Il pulsante "**Vedi mappa**" apre la visualizzazione ad albero del percorso, si può utilizzare anche per spostarsi nella struttura, vedi il paragrafo successivo.

Mappa del percorso



(<https://docs.ai-know.pro/choicemap/img/navigator-mappa.png>)

La mappa mostra l'intero scenario come un albero di nodi collegati. È una visualizzazione che permette di vedere dove ci si trova rispetto alla struttura complessiva.

Ogni nodo è rappresentato da un rettangolo con il titolo abbreviato. I nodi sono colorati in modo diverso in base al loro stato.

Il nodo corrente, quello in cui l'utente si trova, è evidenziato con il colore primario. I nodi già visitati nel percorso corrente hanno un colore che indica "già visto". I nodi non ancora raggiunti sono grigi o con bordo tratteggiato.

Le linee che collegano i nodi rappresentano le scelte. Anche le linee cambiano colore: quelle percorse sono continue ed evidenziate, quelle non ancora esplorate sono tratteggiate. Le frecce poste sulle linee indicano da quale nodo a quale nodo si muove l'utente, i collegamenti infatti sono sempre monodirezionali.

Cliccando su un nodo nella mappa si salta direttamente a quella scheda, se già visitata.

La mappa si chiude con il pulsante di chiusura in alto a destra.

Responsività

L'interfaccia si adatta automaticamente alle dimensioni dello schermo. Su dispositivi mobili i pulsanti diventano più grandi per facilitare il tocco, la mappa si ridimensiona per restare leggibile, il testo mantiene proporzioni adeguate alla lettura.

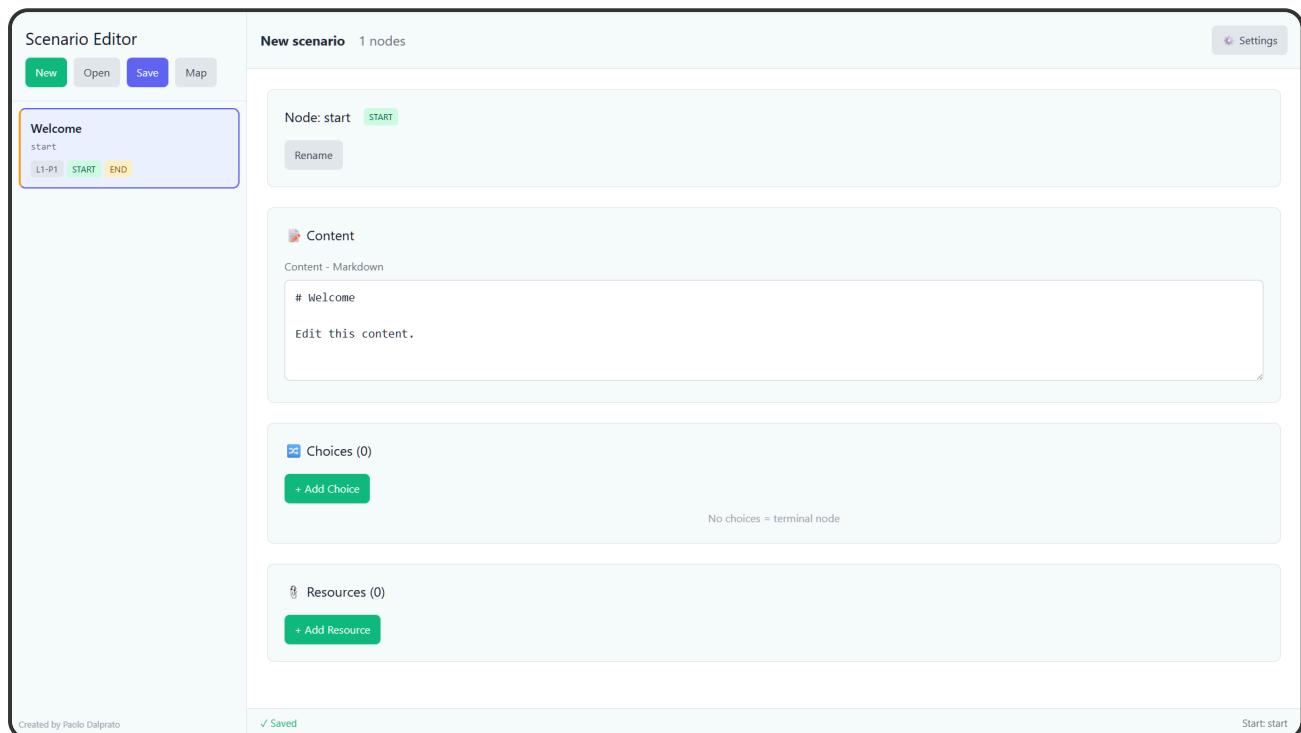
La navigazione rimane funzionale anche su schermi piccoli: tutti i controlli restano accessibili e il contenuto scorre verticalmente quando necessario.

Editor dello scenario

L'editor dello scenario è lo strumento per creare e modificare scenari.

Come si presenta

Uno scenario è l'insieme di nodi (che l'utente nel Navigator vede come schede) e di collegamenti fra i nodi (che l'utente nel Navigator vede come i tasti di scelta all'interno delle schede). Gli scenari sono salvati come file JSON.

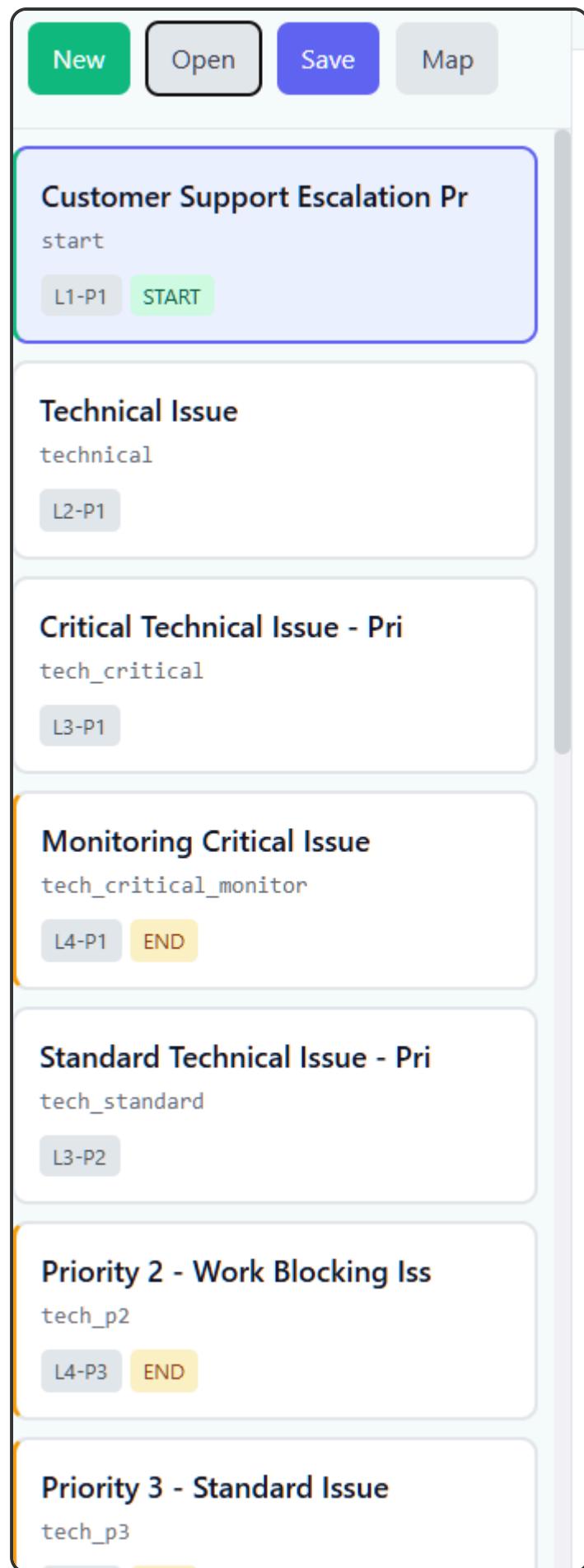


(https://docs.ai-know.pro/choicemap/img/scenario_edit_benvenuto.png)

L'interfaccia è divisa in due parti.

Colonna di sinistra

La colonna di sinistra è dedicata alla gestione delle operazioni generali e dello scenario aperto.



Nella parte alta mostra una riga con i tasti per:

- **New**: creare un nuovo scenario
- **Open**: aprire uno scenario esistente
- **Save**: salvare lo scenario corrente come file JSON
- **Map**: attivare la visualizzazione della mappa

I nomi dei tasti sono nell'inglese di default, ma possono essere personalizzati tramite Settings.

Sotto la riga dei tasti compare l'elenco dei nodi che fanno parte dello scenario aperto: solo il nodo Start se si sta partendo con un nuovo scenario, una lista completa se si è già avanti con la creazione o se si apre uno scenario esistente.

Ogni nodo nella lista mostra:

- il titolo (estratto dal contenuto)
- l'identificativo tecnico
- badge informativi: livello e posizione (es. L2-P1), START per il nodo iniziale, END per i nodi terminali

Area di editing

La parte destra è dedicata all'editing del nodo selezionato.

The screenshot shows a software interface for managing scenarios. At the top, there's a header bar with the title "scenario-sample.json" and "11 nodes". On the right side of the header is a "Settings" button. Below the header, the main workspace is divided into several sections:

- Node: knowledge_theory**: This section contains buttons for "Rename" and "DELETE".
- Content**: A "Content - Markdown" editor with the following text:

```
# Theoretical Understanding
You prefer to build a **mental framework** first.

Theory provides:
```
- Choices (3)**: This section lists three choices for the node:
 - Choice 1**:
 - Button Text: Systematic analysis
 - Target Node: theory_systematic
 - Choice 2**:
 - Button Text: Intuitive synthesis
 - Target Node: theory_intuitive

At the bottom of each choice section is a green "Save Choice" button.

(https://docs.ai-know.pro/choicemap/img/scenario_area_edit.png)

La riga in alto contiene:

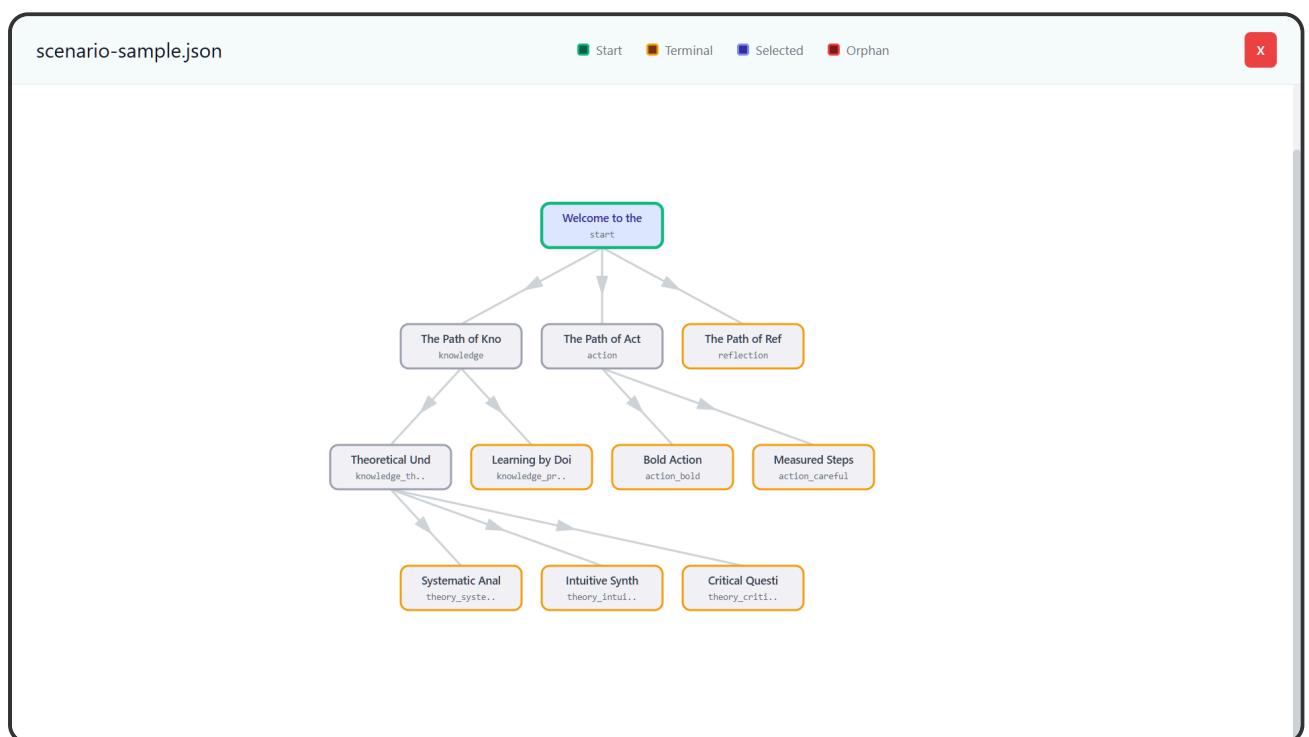
- sulla sinistra: il nome del file dello scenario e il totale dei nodi. Per un nuovo scenario compare "New scenario", il nome cambia automaticamente dopo il salvataggio
- sulla destra: il pulsante **Settings** per i metadati e le traduzioni

Sotto la riga di intestazione compaiono i blocchi che definiscono il nodo:

- **Node**: gestione dell'identificativo del nodo
- **Content**: il contenuto testuale
- **Choices**: le scelte disponibili all'utente
- **Resources**: gli allegati e risorse collegate

Vista mappa

La vista mappa mostra lo scenario come un albero visuale.



(https://docs.ai-know.pro/choicemap/img/scenario_mappa.png)

Ogni nodo mostra il suo identificativo e il titolo. I collegamenti tra nodi rappresentano le scelte disponibili, le frecce indicano la direzione del percorso. Il layout automatico dispone i nodi su livelli, con il nodo iniziale in alto e i rami che si espandono verso il basso.

Codice colori dei bordi:

- **verde**: nodo di partenza (START)
- **arancione**: nodo terminale (END)

- **rosso**: nodo orfano, non raggiungibile
- **viola**: nodo selezionato

I collegamenti all'indietro (loop) sono colorati in arancione per distinguerli dai collegamenti in avanti.

Cliccando su un nodo nella mappa si apre direttamente nell'area di editing. Per chiudere la mappa, cliccare sul pulsante X in alto a destra.

I blocchi dell'area di editing

Node

Il blocco Node gestisce l'identificativo del nodo. Contiene:

- **Rename**: apre un campo per modificare l'identificativo. L'ID deve essere univoco nell'intero scenario e può contenere solo lettere, numeri e underscore
- **DELETE**: elimina l'intero nodo dallo scenario. Non è disponibile per il nodo START. Prima di eliminare, il sistema chiede conferma



Attenzione all'eliminazione

Eliminando un nodo che ha figli collegati, quei figli diventano orfani. È buona pratica eliminare i nodi partendo dalle foglie verso la radice.

Content

Il blocco Content contiene il testo che l'utente vedrà nella scheda del Navigator.

Il contenuto usa una sintassi Markdown semplificata:

Elemento	Sintassi
Titolo grande	# Testo
Titolo	## Testo
Grassetto	**testo**
Corsivo	*testo*
Elemento lista	- testo
Citazione	> testo

Choices

Il blocco Choices definisce le scelte disponibili all'utente, cioè i pulsanti che permettono di procedere nel percorso.

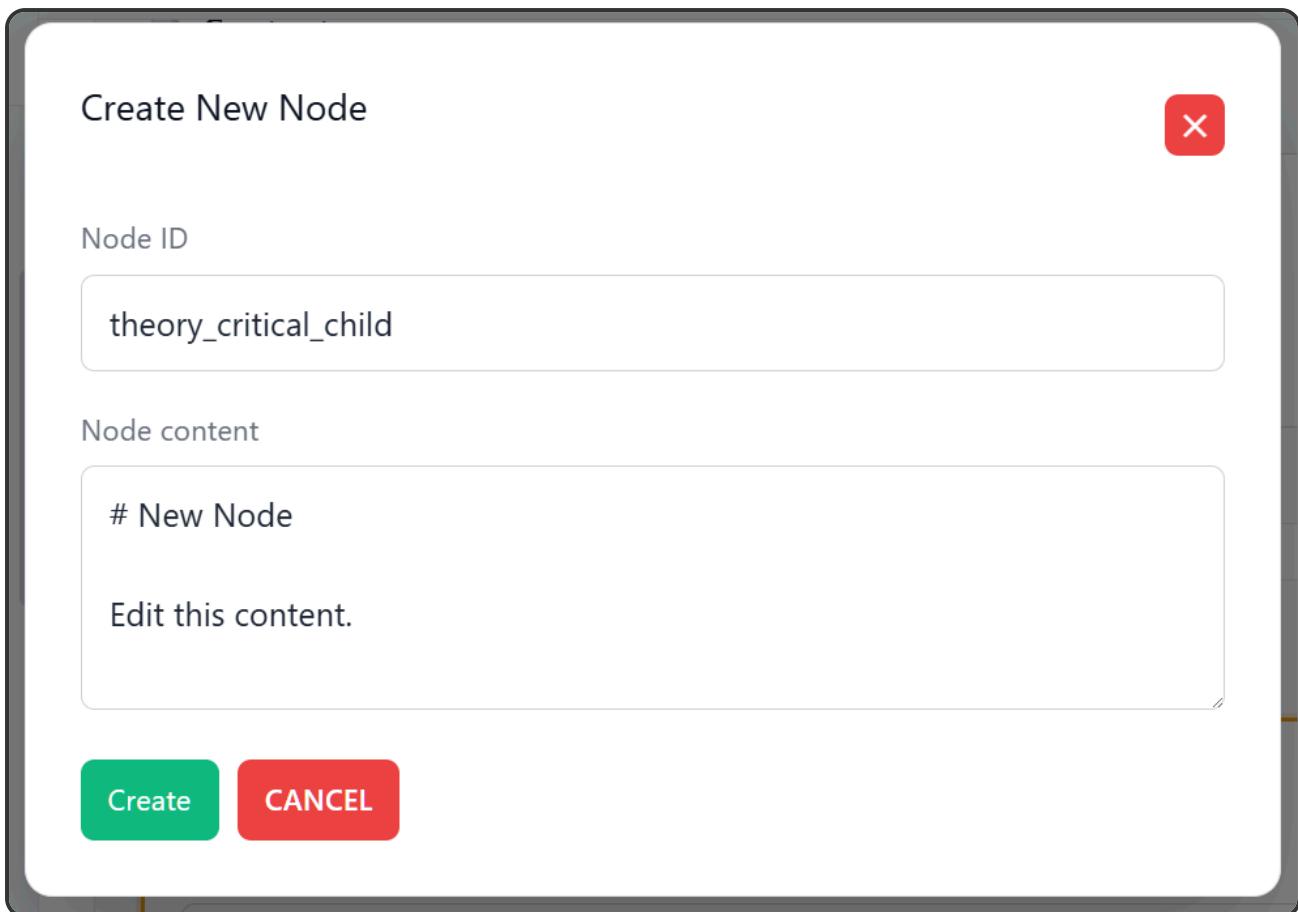
The screenshot shows a modal window titled "Choices (1)". It contains a single choice entry for "Choice 1". The "Button Text" field is labeled "Enter button text" and has a placeholder "Enter button text". The "Target Node" field is labeled "Target Node" and has a dropdown menu with the option "... Select target ...". At the bottom right of the entry is a red "REMOVE" button. Below the entries is a green "Save Choice" button. At the bottom left is a green "+ Add Choice" button.

(https://docs.ai-know.pro/choicemap/img/scenario_add_choice.png)

Cliccando su **+ Add Choice** si aggiunge una nuova scelta con due campi obbligatori:

- **Button Text:** il testo che appare sul pulsante
- **Target Node:** il nodo di destinazione, selezionabile da un menu a tendina che elenca tutti i nodi esistenti

Il menu Target Node include anche l'opzione **+ Create new node** che apre un popup per creare direttamente un nuovo nodo destinazione.



(https://docs.ai-know.pro/choicemap/img/scenario_new_node.png)

Indicatori visivi di stato:

- bordo grigio: scelta appena creata, non ancora compilata
- bordo ambra con "● unsaved": modifiche non salvate
- bordo verde con "✓": scelta completa e salvata
- bordo rosso: errori di validazione

Il pulsante **Save Choice** salva la scelta solo se entrambi i campi sono compilati. Il pulsante **REMOVE** elimina la scelta.

Un nodo può avere più scelte che portano a nodi diversi. È possibile creare loop collegando un nodo a uno dei suoi antenati.

Resources

Il blocco Resources permette di allegare risorse esterne al nodo.

The screenshot shows a user interface for managing resources. At the top, there's a header with a gear icon and the text 'Resources (1)'. Below it, a card for 'Resource 1' is displayed. The card has fields for 'Type *' (set to 'Link'), 'Label *' (set to 'Resource description'), and 'URL *' (containing 'https://...'). A red 'REMOVE' button is in the top right corner of the card. At the bottom of the card is a green 'Save Resource' button. Below the card is a green button labeled '+ Add Resource'.

(https://docs.ai-know.pro/choicemap/img/scenario_new_resource.png)

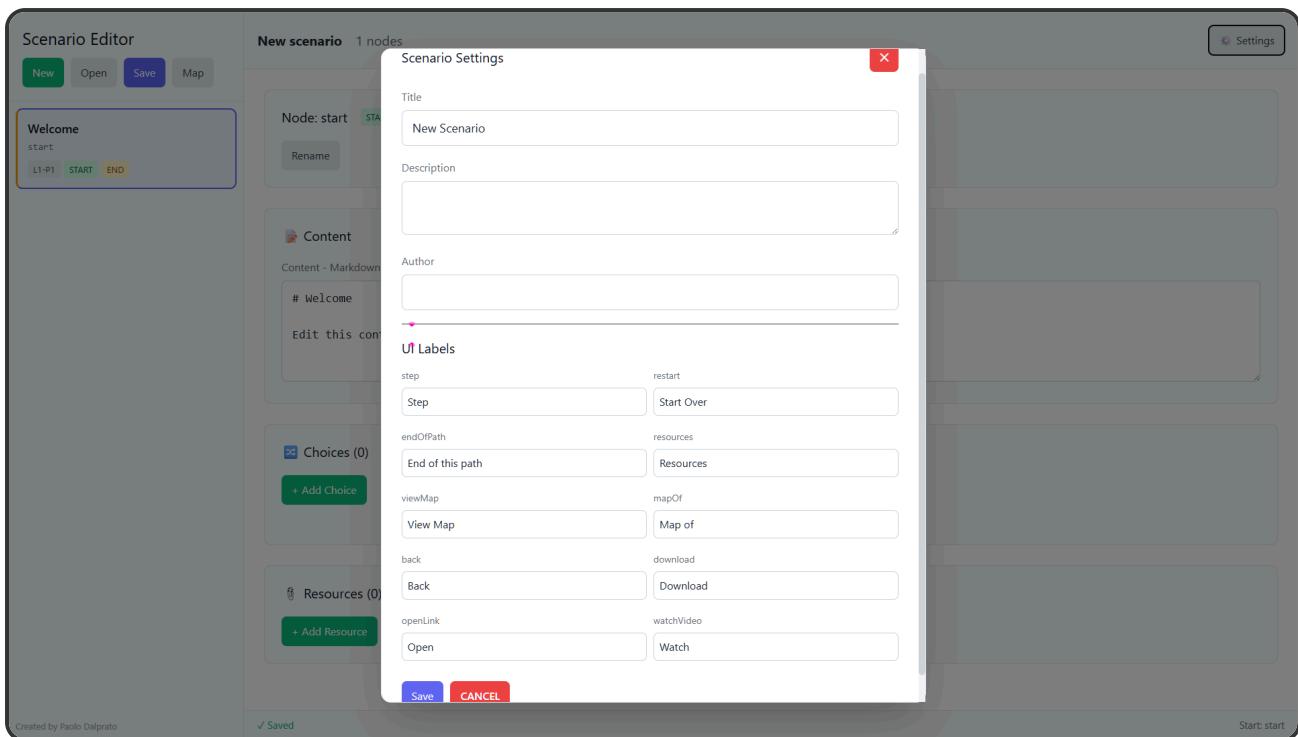
Cliccando su **+ Add Resource** si aggiunge una risorsa con tre campi obbligatori:

- **Type:** il tipo di risorsa
 - *Link*: collegamento a una pagina web
 - *Download*: documento scaricabile (deve essere accessibile via URL)
 - *Video*: video YouTube (usare il link di condivisione)
- **URL**: l'indirizzo della risorsa, può essere sia un link web sia un url relativo ad una risorsa locale. In questo caso conviene definire una sottocartella in cui raccogliere gli elementi, l'url potrebbe essere "**docs/elemento1.pdf**"
- **Label**: la descrizione che appare all'utente nel Navigator

Gli indicatori visivi di stato funzionano come per le Choices. Il pulsante **Save Resource** salva solo se tutti i campi sono compilati.

Settings: metadati e traduzioni

Il pulsante Settings apre un pannello per configurare i metadati dello scenario e le etichette dell'interfaccia.



(https://docs.ai-know.pro/choicemap/img/scenario_metadata.png)

Metadati:

- **Title:** il nome dello scenario, visibile nell'intestazione del Navigator
- **Description:** descrizione breve (opzionale)
- **Author:** nome dell'autore (opzionale)

UI Labels: permette di tradurre o personalizzare tutte le etichette dell'interfaccia utente (step, restart, endOfPath, resources, viewMap, back, download, openLink, watchVideo).

Il file JSON

Uno scenario viene salvato come file JSON strutturato in tre sezioni:

(https://docs.ai-know.pro/choicemap/img/scenario_json.png)

- **meta**: i metadati (title, description, author)
 - **translations**: le etichette personalizzate
 - **startNode**: l'identificativo del nodo di partenza
 - **nodes**: l'oggetto contenente tutti i nodi con i loro contenuti, scelte e risorse

Salvataggio

Il pulsante Save genera un file JSON. Non c'è salvataggio automatico: salvare regolarmente durante il lavoro.

Nella barra di stato in basso è sempre visibile lo stato:

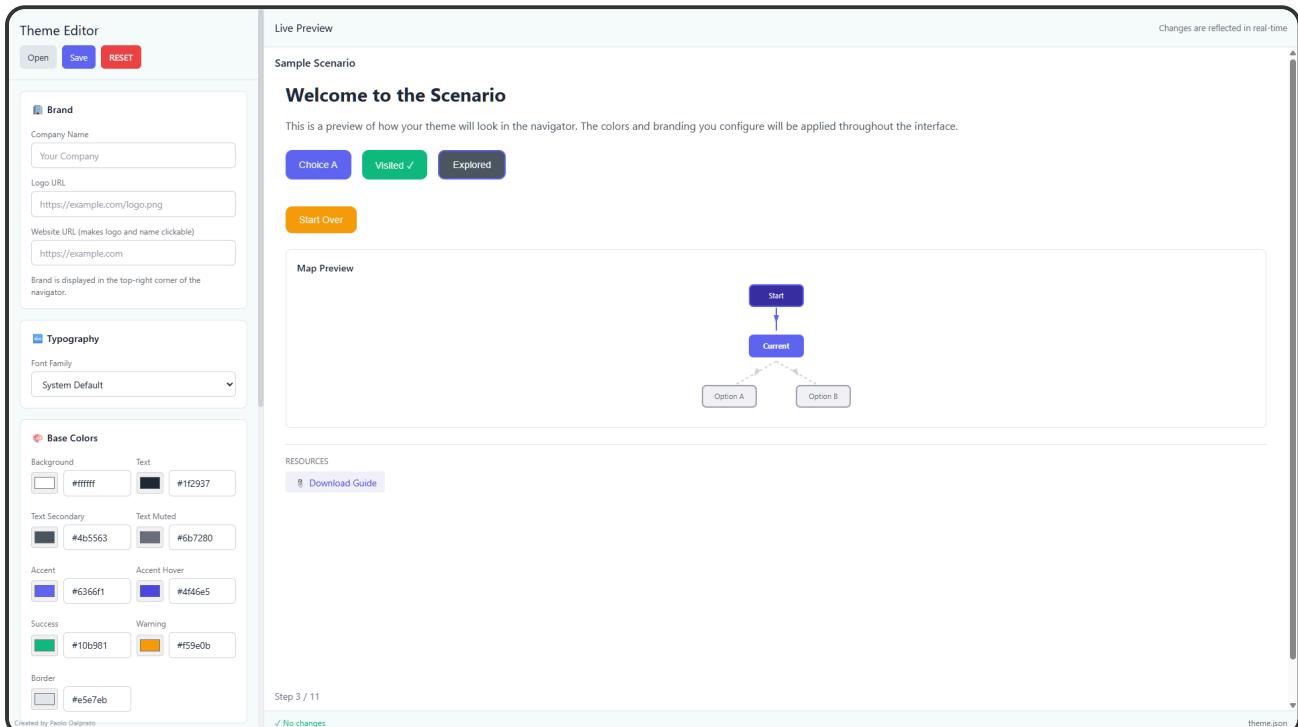
- "● Unsaved" (arancione): ci sono modifiche non salvate
 - "✓ Saved" (verde): tutto salvato

Interfaccia del Theme Editor

Il Theme Editor permette di personalizzare l'aspetto visivo del Navigator senza modificare codice. Ogni modifica si riflette immediatamente in un'anteprima, permettendo di vedere il risultato finale mentre si lavora. Il tema viene salvato in un file JSON.

Layout generale

L'interfaccia è divisa in due aree principali affiancate.



(https://docs.ai-know.pro/choicemap/img/theme_general.png)

Colonna di sinistra

Nella colonna di sinistra, in cima ci sono i tasti per le azioni generali:

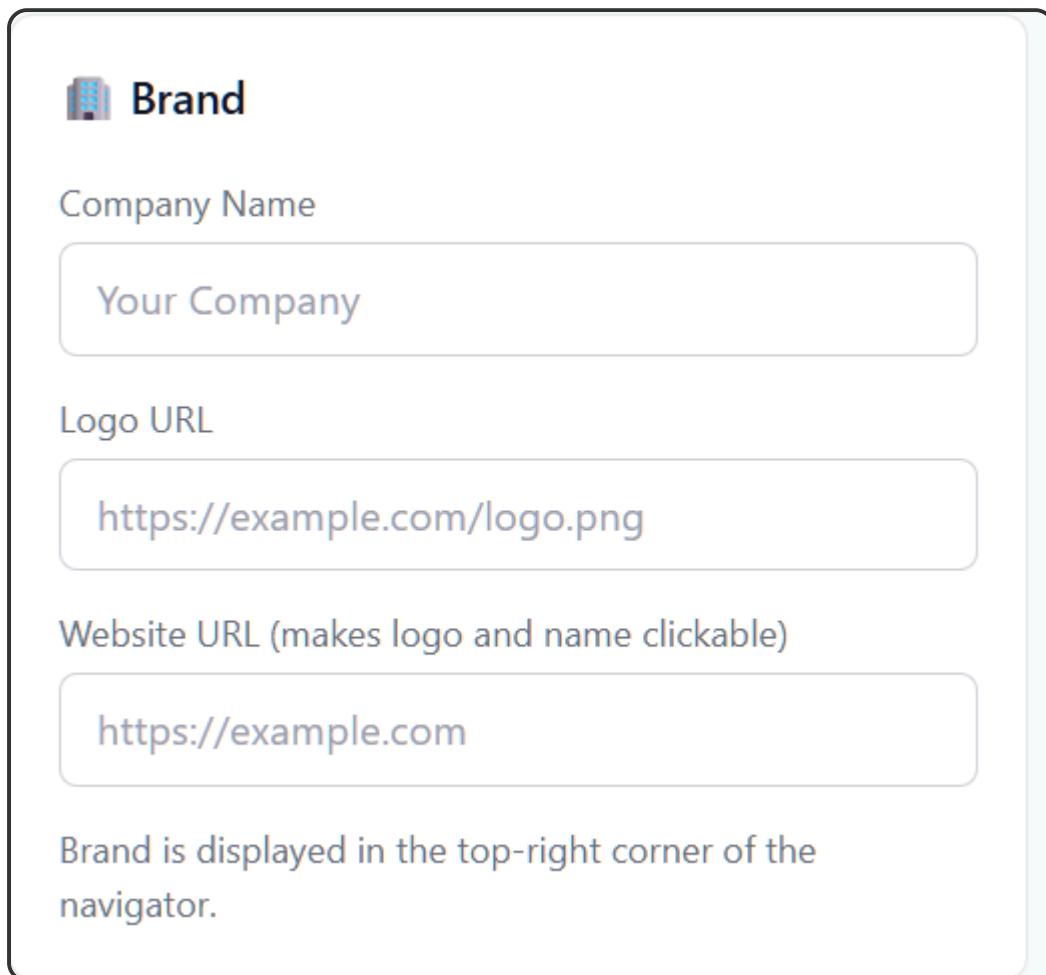
- **Open:** per aprire il file di un tema.
- **Save:** per salvare il tema in un file.
- **RESET:** per riportare il tema ai valori di default.

Sotto i tasti ci sono i pannelli di controllo organizzati in sezioni espandibili, ogni sezione raggruppa impostazioni correlate. Le sezioni sono:

- **Brand**: raccoglie gli elementi di branding aziendale.
- **Typography**: definisce il font da usare.
- **Colori**: definisce i colori dei vari elementi grafici, è suddivisa in blocchi per tipologia di elementi.

Sezione brand

La sezione brand controlla gli elementi di identità aziendale che appaiono nel Navigator nell'angolo destro in alto.

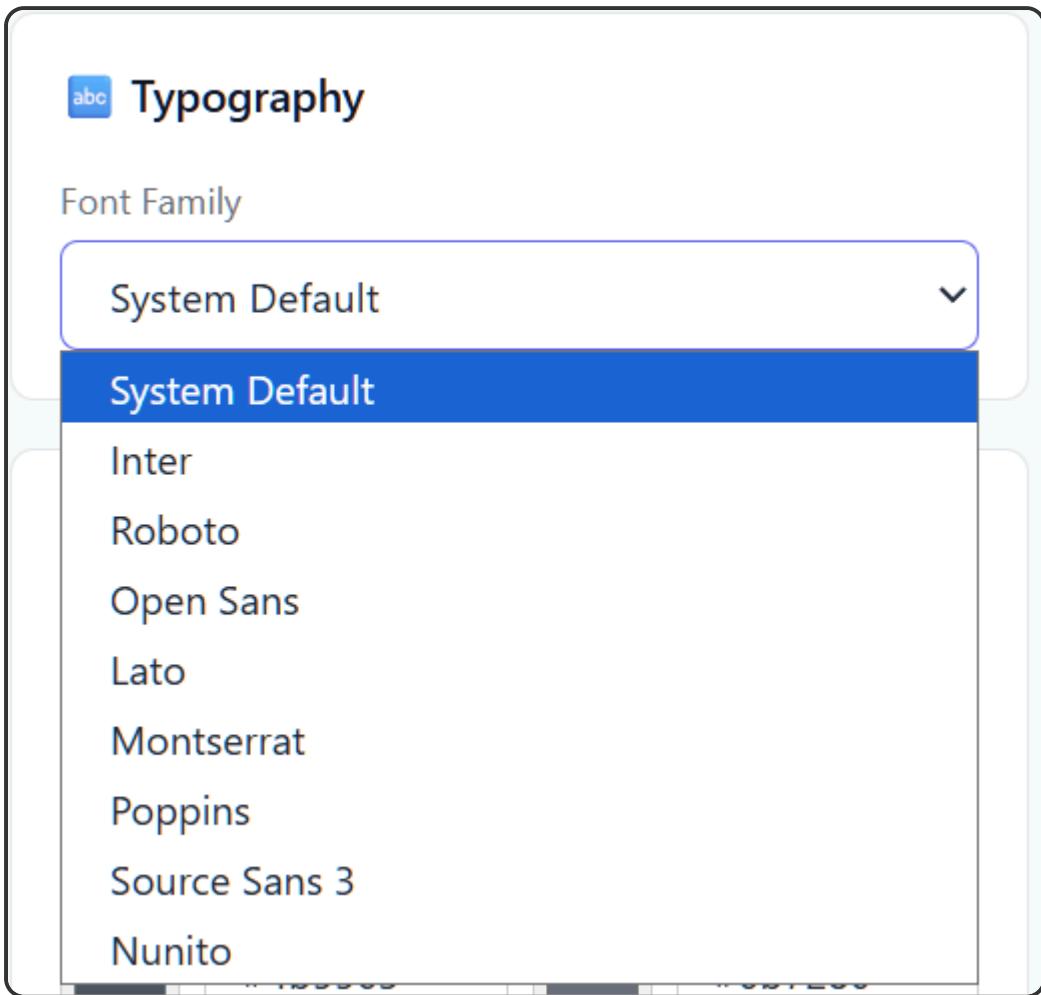


(https://docs.ai-know.pro/choicemap/img/theme_brand.png)

- **Company Name**: il campo per il nome dell'azienda permette di inserire un testo che appare accanto o sotto il logo. Se non si vuole mostrare il nome, basta lasciare il campo vuoto.
- **Logo URL**: il campo logo accetta un URL che punta all'immagine del logo. Può essere un URL esterno o un percorso relativo a un file nella cartella del progetto o, meglio, ad una sottocartella. Le dimensioni vengono adattate automaticamente per non interferire con il contenuto.
- **Website URL**: l'URL del sito web rende cliccabile il logo e il nome azienda (se presente). L'utente può cliccare per visitare il sito in una nuova scheda.

Sezione typography

La sezione tipografia controlla il font usato in tutta l'interfaccia.



(https://docs.ai-know.pro/choicemap/img/theme_typography.png)

Un menu a tendina presenta le opzioni disponibili. La prima opzione è "System Default", che usa il font di sistema del dispositivo dell'utente senza caricare font esterni.

Le altre opzioni sono font Google Fonts: - Inter - Roboto - Open Sans - Lato - Montserrat - Poppins - Source Sans 3 - Nunito

Selezionando uno di questi, il font viene caricato automaticamente da Google e applicato all'interfaccia.

L'anteprima mostra immediatamente l'effetto del cambio di font su titoli, testo e pulsanti.

Sezione colori

La sezione colori è la più articolata, con controlli per ogni elemento dell'interfaccia.

Base Colors

Background



#ffffff

Text



#1f2937

Text Secondary



#4b5563

Text Muted



#6b7280

Accent



#6366f1

Accent Hover



#4f46e5

Success



#10b981

Warning



#f59e0b

Border



#e5e7eb

Buttons

Choice Background



#6366f1

Choice Text



#ffffff

Visited Background



#10b981

Visited Text



#ffffff

Explored Background



#4b5563

Explored Text



#ffffff

Restart Background



#f59e0b

Restart Text



#ffffff

Map

Current Node

Created by Paolo Dalprato

Current Text

- **Base Colors:** i colori generali definiscono lo sfondo della pagina, il colore del testo principale, il colore del testo secondario (usato per informazioni meno importanti), il colore accent (usato per evidenziare elementi interattivi). Ogni controllo è un selettore colore che apre un pannello per scegliere la tonalità desiderata. Il valore esadecimale è visibile e modificabile direttamente per chi preferisce inserire codici colore precisi.
- **Buttons:** i pulsanti di scelta hanno controlli separati per ogni stato: colore di sfondo e colore del testo per le scelte non ancora esplorate, per le scelte già fatte, per le scelte esplorate ma non nel percorso corrente. Questa distinzione cromatica aiuta l'utente a capire cosa ha già visto quando torna indietro per esplorare percorsi alternativi. Il pulsante di restart ha colori propri, tipicamente diversi dai pulsanti di scelta per distinguere visivamente.
- **Map:** il blocco mappa controlla i colori della visualizzazione ad albero: il nodo corrente, i nodi visitati, i nodi non ancora raggiunti, le linee di collegamento. Questi colori dovrebbero essere coerenti con i colori dei pulsanti per mantenere un linguaggio visivo consistente.

Anteprima live

L'area di anteprima sulla destra mostra un Navigator funzionante con uno scenario ed una mappa di esempio. Ogni modifica ai controlli si riflette istantaneamente nell'anteprima.

L'anteprima usa uno scenario semplice predefinito. Per vedere come il tema funziona con lo scenario reale, è necessario salvare il tema e caricarlo nel Navigator vero.

Flusso di lavoro tipico

Il flusso tipico prevede di partire da un tema esistente o dai valori predefiniti, modificare le impostazioni verificando l'effetto nell'anteprima, salvare il risultato come file JSON.

Il file tema salvato va poi indicato nel file di configurazione `config.json` del progetto, associandolo allo scenario desiderato.

È possibile creare più temi diversi per lo stesso scenario, ad esempio una versione chiara e una scura, o temi personalizzati per diversi clienti.

Creare uno scenario

Questa guida accompagna passo passo nella creazione di uno scenario completo, dalla pianificazione alla pubblicazione. Per i dettagli sull'interfaccia dell'editor, vedi [Interfaccia dello Scenario Editor](https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/) (<https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/>).

Pianificare prima di costruire

Prima di aprire l'editor, è utile avere chiara la struttura del percorso:

- Qual è il punto di partenza?
- Quali sono i punti decisionali chiave?
- Quali sono i possibili finali?
- Ci sono percorsi che si ricongiungono o loop?

Uno schema su carta o una mappa mentale aiutano a visualizzare la struttura. Non serve che sia perfetto: lo scenario evolverà durante la creazione, ma partire con un'idea chiara evita di perdersi.

Workflow consigliato

1. Configurare i metadati

Aprire `scenario-editor.html` nel browser. L'editor si presenta con un nuovo scenario vuoto contenente solo il nodo Start.

Prima di lavorare sui nodi, impostare i metadati cliccando su **Settings**:

- **Title**: il nome che apparirà nell'intestazione del Navigator
- **Description**: una breve descrizione (opzionale, utile per ricordare lo scopo)
- **Author**: il nome dell'autore (opzionale)

Se lo scenario sarà in una lingua diversa dall'inglese, ad esempio l'italiano, oppure si desidera usare altri testi per le azioni dei tasti, si deve personalizzare anche le **UI Labels** per tradurre i testi dei pulsanti (**Passo** al posto di **Step**, **Ricomincia** al posto di **Step Over**, **Fine del percorso** al posto di **End of this path**, ecc.).

2. Costruire dal nodo iniziale

Partire dal nodo Start e procedere in profondità:

1. Selezionare il nodo nella colonna sinistra
2. Scrivere il contenuto nel blocco **Content** usando la [sintassi Markdown](https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#content) (<https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#content>)
3. Aggiungere le scelte nel blocco **Choices**
4. Per ogni scelta, selezionare un nodo esistente o crearne uno nuovo con **+ Create new node**

Procedere così per tutti i nodi, ramificando la struttura secondo il piano iniziale.



Convenzioni per gli identificativi

Usare nomi brevi e descrittivi senza spazi: `intro`, `domanda_1`, `risposta_ok`, `finale_positivo`. Gli identificativi non sono visibili agli utenti ma aiutano a orientarsi nella struttura.

3. Verificare con la mappa

Periodicamente, usare il pulsante **Map** per visualizzare l'intera struttura. La mappa mostra:

- Il flusso generale dei percorsi
- Eventuali nodi orfani (bordo rosso) da collegare o eliminare
- I nodi terminali (bordo arancione)
- I loop, evidenziati con collegamenti arancioni

Per i dettagli sul codice colori, vedi [Vista mappa](https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#vista-mappa) (<https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#vista-mappa>).

4. Salvare regolarmente

Non c'è salvataggio automatico. Cliccare **Save** frequentemente per evitare di perdere il lavoro. Il pulsante genera un file JSON da scaricare.

La barra di stato in basso mostra sempre se ci sono modifiche non salvate.

Gestire i nodi terminali

Un nodo senza scelte è automaticamente un nodo terminale. Nel Navigator, al posto dei pulsanti appare il messaggio "Fine del percorso" e il pulsante per ricominciare.

- Per creare un terminale: rimuovere tutte le scelte dal nodo
- Per trasformare un terminale in nodo intermedio: aggiungere almeno una scelta

Gestire i nodi orfani

Un nodo orfano esiste nello scenario ma non è raggiungibile dal nodo iniziale attraverso nessun percorso. L'editor li segnala con l'etichetta "orphan" nella lista e il bordo rosso nella mappa.

I nodi orfani si creano tipicamente quando:

- Si elimina un nodo che era l'unico collegamento verso altri nodi
- Si rimuove una scelta che era l'unico percorso verso un nodo

Per risolvere: collegare il nodo aggiungendo una scelta che punta a lui da qualche altro nodo, oppure eliminarlo se non serve più.

Eliminare nodi in sicurezza

Per eliminare un nodo, selezionarlo e cliccare **DELETE** nel blocco Node.

Attenzione

Eliminando un nodo che ha figli, quei figli possono diventare orfani se non hanno altri collegamenti.

Best practice: eliminare i nodi partendo dalle foglie (nodi terminali) e risalendo verso la radice. In questo modo non si creano orfani accidentalmente.

Creare loop

È possibile creare percorsi che tornano indietro collegando un nodo a uno dei suoi antenati. Nella vista mappa, questi collegamenti sono evidenziati in arancione.

Casi d'uso tipici:

- Permettere all'utente di rivedere una sezione
- Creare menu principale a cui tornare
- Gestire errori con "riprova"
- Rappresentare strutture **repeat ... until** (o equivalenti)

Aggiungere risorse

Per allegare documenti, link o video a un nodo, usare il blocco **Resources**. Per i dettagli sui tipi disponibili e la configurazione, vedi [Resources](https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#resources) (<https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#resources>).

Le risorse appaiono nel Navigator come pulsanti sotto il contenuto del nodo.



Risorse locali

Per file locali (PDF, documenti), si consiglia di creare una sottocartella `docs/` e usare percorsi relativi come `docs/manuale.pdf`.

Modificare uno scenario esistente

Per riprendere il lavoro su uno scenario:

1. Cliccare **Open** e selezionare il file JSON
2. Lo scenario si carica con tutti i nodi nella colonna sinistra
3. Selezionare un nodo per modificarlo
4. Salvare le modifiche con **Save**

Dalla creazione alla pubblicazione

Una volta completato lo scenario:

1. Salvare il file JSON con un nome descrittivo (es. `onboarding-nuovi-dipendenti.json`)
2. Copiare il file nella cartella del Navigator
3. Modificare `config.json` per puntare al nuovo scenario
4. Testare aprendo `choicemap.html`

Per i dettagli sulle opzioni di pubblicazione (intranet, GitHub Pages, embedding), vedi [Pubblicazione](#) (<https://docs.ai-know.pro/choicemap/pubblicazione/>).

Struttura dei contenuti

Questa sezione descrive l'architettura dei file e il formato JSON che compone uno scenario. La conoscenza di questa struttura non è necessaria per usare lo Scenario Editor, ma è utile per modifiche avanzate, debugging, o per generare scenari programmaticamente.

Architettura dei file

ChoiceMap separa nettamente il framework (codice) dai contenuti (dati). I file HTML contengono l'applicazione e non vanno modificati; tutte le personalizzazioni avvengono tramite file JSON.

File HTML (framework)

I tre file HTML sono applicazioni complete che non richiedono modifiche:

File	Funzione
choicemap.html	Visualizza gli scenari per l'utente finale
scenario-editor.html	Editor per creare e modificare scenari
theme-editor.html	Editor per personalizzare l'aspetto

Questi file caricano React e Tailwind CSS da CDN e leggono i contenuti dai file JSON. Non è necessario (né consigliato) modificarli.

config.json (il regista)

Il file `config.json` è il punto di ingresso che dice al Navigator cosa caricare:

```
{  
  "scenario": "scenario-mio-progetto.json",  
  "theme": "theme.json",  
  "showCredits": true  
}
```

Campo	Funzione
scenario	Nome del file JSON dello scenario da visualizzare
theme	Nome del file JSON del tema da applicare
showCredits	Mostra/nasconde il footer con i crediti dell'autore

Cambiando il valore di `scenario` e ricaricando il Navigator, si passa a uno scenario diverso senza modificare altro. Lo stesso per `theme`.

defaults.json (valori predefiniti)

Contiene tutti i valori di default per il tema: colori, font, configurazione del branding. È il "tema base" da cui partono tutti gli altri.

```
{
  "brand": {
    "name": "",
    "logo": "",
    "website": ""
  },
  "colors": {
    "background": "#ffffff",
    "text": "#1f2937",
    "accent": "#6366f1"
  }
}
```

Questo file non va modificato. Per personalizzare l'aspetto, si crea un file tema separato.

theme.json (personalizzazioni)

Il file tema contiene solo le personalizzazioni rispetto ai default. Il Navigator legge prima `defaults.json`, poi sovrascrive con i valori presenti in `theme.json`.

Se `theme.json` contiene solo il logo aziendale, tutti gli altri valori (colori, font) vengono presi da `defaults.json`:

```
{
  "brand": {
    "name": "Azienda XYZ",
    "logo": "images/logo.png"
  }
}
```

Questo approccio permette di creare temi minimali che specificano solo ciò che cambia.

shared-styles.css (solo per gli editor)

Contiene stili CSS condivisi tra Scenario Editor e Theme Editor. Non viene usato dal Navigator e non serve includerlo quando si pubblica uno scenario per gli utenti finali.

Il file scenario

Ogni scenario è un file JSON con una struttura ben definita:

```
{
  "meta": {
    "title": "Titolo dello scenario",
    "description": "Descrizione opzionale",
    "author": "Nome autore"
  },
  "translations": {
    "step": "Passo",
    "restart": "Ricomincia",
    "endOfPath": "Fine del percorso",
    "resources": "Risorse",
    "viewMap": "Vedi mappa",
    "mapOf": "Mappa di",
    "back": "Indietro",
    "download": "Scarica",
    "openLink": "Apri",
    "watchVideo": "Guarda"
  },
  "startNode": "start",
  "nodes": {
    "start": {
      "content": "# Benvenuto\n\nContenuto in **Markdown**.",
      "choices": [
        { "text": "Prima opzione", "next": "nodo_a" },
        { "text": "Seconda opzione", "next": "nodo_b" }
      ],
      "resources": []
    }
  }
}
```

Sezione meta

La sezione `meta` contiene le informazioni descrittive dello scenario.

Campo	Obbligatorio	Uso
<code>title</code>	Sì	Appare nell'intestazione del Navigator
<code>description</code>	No	Note per chi gestisce lo scenario
<code>author</code>	No	Mostrato nel footer se <code>showCredits</code> è attivo

Sezione translations

Definisce le etichette dell'interfaccia, permettendo di localizzare lo scenario in qualsiasi lingua.

Chiave	Uso	Esempio italiano
<code>step</code>	Indicatore di progresso	Passo
<code>restart</code>	Pulsante ricomincia	Ricomincia
<code>endOfPath</code>	Messaggio fine percorso	Fine del percorso
<code>resources</code>	Titolo sezione risorse	Risorse
<code>viewMap</code>	Pulsante apri mappa	Vedi mappa
<code>mapof</code>	Titolo della mappa	Mappa di
<code>back</code>	Pulsante indietro	Indietro
<code>download</code>	Etichetta download	Scarica
<code>openLink</code>	Etichetta link	Apri
<code>watchVideo</code>	Etichetta video	Guarda

Per un quiz, si potrebbe usare "Domanda" invece di "Passo" e "Riprova" invece di "Ricomincia".

Campo startNode

Indica quale nodo è il punto di partenza dello scenario. Il valore deve corrispondere a una chiave presente nell'oggetto `nodes`.

Tipicamente si usa "start" per convenzione, ma può essere qualsiasi identificativo valido.

Sezione nodes

Un oggetto dove ogni chiave è l'identificativo di un nodo e il valore contiene i dati del nodo.

Identificativo del nodo: la chiave di ogni nodo è il suo ID univoco, usato internamente per i collegamenti.

Convenzioni consigliate: - Solo lettere minuscole, numeri e underscore - Nomi descrittivi ma brevi: `intro`, `domanda_1`, `risposta_ok` - Evitare spazi e caratteri speciali

Contenuto del nodo: il campo `content` contiene il testo in formato Markdown. Per la sintassi supportata, vedi [Content](https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#content) (<https://docs.ai-know.pro/choicemap/interfaccia-scenario-editor/#content>).

Per andare a capo nel JSON, usare `\n`:

```
"content": "Prima riga\n\nSeconda riga dopo uno spazio"
```

Scelte del nodo: il campo `choices` è un array di oggetti che definiscono i pulsanti di scelta:

```
"choices": [
    { "text": "Testo del pulsante", "next": "id_nodo_destinazione" }
]
```

- `text`: quello che l'utente legge sul pulsante
- `next`: l'identificativo del nodo di destinazione

Un nodo senza scelte (array vuoto `[]`) è un nodo finale.

Risorse del nodo: il campo `resources` è opzionale e definisce allegati:

```
"resources": [
    { "type": "download", "label": "Scarica il PDF", "url": "docs/guida.pdf" },
    { "type": "link", "label": "Approfondisci", "url": "https://esempio.com" },
    { "type": "video", "label": "Video tutorial", "url": "https://youtube.com/watch?v=..." }
]
```

Tipo	Comportamento
<code>download</code>	Scarica il file
<code>link</code>	Apre in nuova scheda
<code>video</code>	Apre in nuova scheda

Validazione

Il Navigator mostra errori se qualcosa non funziona nei file JSON. Errori comuni:

- JSON malformato (virgole mancanti, virgolette non chiuse)
- `startNode` che punta a un nodo inesistente
- Scelte che puntano a nodi inesistenti
- Sezione `translations` mancante o incompleta

Lo Scenario Editor genera sempre JSON valido, ma modifiche manuali possono introdurre errori. In caso di problemi, verificare la sintassi con un validatore JSON online.

Personalizzare il tema

Questa guida accompagna nella personalizzazione dell'aspetto visivo del Navigator. Per i dettagli sull'interfaccia dell'editor, vedi [Interfaccia del Theme Editor](https://docs.ai-know.pro/choicemap/interfaccia-theme-editor/) (<https://docs.ai-know.pro/choicemap/interfaccia-theme-editor/>).

Workflow consigliato

1. Aprire il Theme Editor (<http://localhost:8000/theme-editor.html>)
2. Partire dal tema predefinito o caricarne uno esistente con **Open**
3. Modificare le impostazioni verificando l'effetto nell'anteprima
4. Salvare con **Save** e copiare il file nella cartella del progetto
5. Aggiornare `config.json` per puntare al nuovo tema
6. Verificare nel Navigator reale con uno scenario completo

L'anteprima del Theme Editor usa uno scenario semplificato. Per vedere il risultato finale con contenuti reali, è necessario testare nel Navigator.

Consigli per il branding

Logo

Formato	Uso consigliato
SVG	Scalabilità ottimale, file leggero
PNG trasparente	Buona qualità, compatibilità universale
JPG	Solo se necessario, evitare per loghi

Evitare immagini troppo grandi: il logo viene ridimensionato automaticamente ma file pesanti rallentano il caricamento. Dimensione consigliata: massimo 200x60 pixel o equivalente.

Il campo accetta sia URL esterni (<https://azienda.com/logo.png>) sia percorsi relativi (`images/logo.png`). Per progetti distribuibili, preferire percorsi relativi.

Consigli per la tipografia

Contesto	Font consigliati
Aziendale/professionale	Inter, Roboto, Source Sans 3
Creativo/informale	Poppins, Nunito, Montserrat
Massima compatibilità	System Default

"System Default" non carica font esterni: usa quello del dispositivo dell'utente. È la scelta più leggera e garantisce leggibilità su qualsiasi sistema.

Consigli per i colori

Contrasto

Il testo deve essere leggibile sullo sfondo. Combinazioni sicure:

Sfondo	Testo principale	Testo secondario
Bianco (#ffffff)	Grigio scuro (#1f2937)	Grigio medio (#4b5563)
Grigio scuro (#1f2937)	Bianco (#ffffff)	Grigio chiaro (#9ca3af)

Colore accent

Il colore accent definisce la personalità dell'interfaccia. Usare il colore principale del brand aziendale per coerenza visiva. Verificare che sia sufficientemente saturo da risaltare come elemento interattivo.

Stati dei pulsanti

La distinzione cromatica tra pulsanti aiuta l'utente a orientarsi:

- **Scelte disponibili:** colore accent, invita all'azione
- **Scelte già fatte:** colore diverso (verde predefinito), conferma visiva
- **Scelte esplorate:** grigio neutro, indica "già visto in un altro percorso"

Mantenere coerenza tra i colori dei pulsanti e quelli della mappa per un linguaggio visivo uniforme.

Salvare e applicare

Il pulsante **Save** genera un file JSON. Rinominarlo con un nome descrittivo:

```
theme-azienda-2024.json  
theme-corso-sicurezza.json  
theme-dark.json
```

Per applicare il tema, aggiornare `config.json`:

```
{  
  "scenario": "scenario-corso.json",  
  "theme": "theme-azienda-2024.json",  
  "showCredits": false  
}
```

Ricaricare il Navigator (Ctrl+Shift+R) per vedere il tema applicato.

Gestire temi multipli

È possibile creare più temi per contesti diversi:

- **Varianti di luminosità:** tema chiaro e tema scuro
- **Clienti diversi:** un tema per ogni cliente con il suo branding
- **Contesti diversi:** tema formale per compliance, tema vivace per onboarding

Cambiando il valore `theme` in `config.json` si passa da un tema all'altro senza modificare lo scenario. Lo stesso contenuto può avere aspetti completamente diversi.

Modificare il tema via JSON

Per chi preferisce lavorare direttamente sul file, la struttura è:

```
{
  "brand": {
    "name": "Nome Azienda",
    "logo": "images/logo.png",
    "website": "https://esempio.com"
  },
  "typography": {
    "fontFamily": "inter"
  },
  "colors": {
    "background": "#ffffff",
    "text": "#1f2937",
    "textSecondary": "#4b5563",
    "accent": "#6366f1"
  },
  "buttons": {
    "choiceBackground": "#6366f1",
    "choiceText": "#ffffff",
    "visitedBackground": "#10b981",
    "visitedText": "#ffffff",
    "exploredBackground": "#e5e7eb",
    "exploredText": "#374151",
    "restartBackground": "#f59e0b",
    "restartText": "#ffffff"
  },
  "map": {
    "nodeCurrent": "#6366f1",
    "nodeCurrentText": "#ffffff",
    "nodeVisited": "#3730a3",
    "nodeVisitedText": "#ffffff",
    "nodeUnvisited": "#f3f4f6",
    "nodeUnvisitedText": "#6b7280",
    "nodeUnvisitedBorder": "#d1d5db",
    "lineVisited": "#6366f1",
    "lineUnvisited": "#d1d5db"
  }
}
```

Tutti i campi sono opzionali. I valori mancanti vengono presi da `defaults.json`. Questo permette di creare temi minimali che specificano solo ciò che cambia:

```
{
  "brand": {
    "name": "Azienda XYZ",
    "logo": "images/logo-xyz.png"
  },
  "colors": {
    "accent": "#dc2626"
  }
}
```

Valori disponibili per `fontFamily`: `system`, `inter`, `roboto`, `open-sans`, `lato`, `montserrat`, `poppins`, `source-sans-3`, `nunito`.

Pubblicazione

Questa sezione copre l'organizzazione del lavoro, i ruoli coinvolti e le opzioni per rendere gli scenari accessibili agli utenti finali.

Ruoli e responsabilità

In un contesto aziendale, due figure distinte interagiscono con ChoiceMap.

Utente finale

L'utente finale utilizza **solo il Navigator** per fruire degli scenari. Non ha accesso agli editor e non modifica i contenuti.

Aspetto	Dettaglio
Usa	Solo <code>choicemap.html</code>
Accede a	Cartella di pubblicazione con file minimi
Competenze richieste	Nessuna, solo aprire un link nel browser
File visibili	Navigator + config + scenario + tema

Amministratore

L'amministratore crea, modifica e pubblica gli scenari. Ha accesso completo a tutti gli strumenti.

Aspetto	Dettaglio
Usa	Navigator + Scenario Editor + Theme Editor
Accede a	Repository di lavoro completo
Competenze richieste	Gestione file, comprensione della struttura
File gestiti	Tutti i file del progetto

In contesti piccoli, la stessa persona può coprire entrambi i ruoli. In organizzazioni più grandi, i ruoli sono tipicamente separati.

Workflow dell'amministratore

Repository di lavoro

L'amministratore mantiene un repository locale (una cartella sul proprio computer o su un server) con tutti i file del progetto:

```
choicemap/
└── choicemap.html      # Navigator
    ├── scenario-editor.html      # Editor scenari
    ├── theme-editor.html        # Editor temi
    ├── config.json            # Configurazione
    ├── defaults.json          # Valori predefiniti
    ├── shared-styles.css       # Stili condivisi
    └── scenari/
        ├── onboarding-v1.json
        ├── onboarding-v2.json
        └── compliance-draft.json
    └── temi/
        └── tema-aziendale.json
    └── risorse/
        ├── manuale.pdf
        └── checklist.docx
```

Ciclo di lavoro tipico

1. **Creare o modificare lo scenario** usando `scenario-editor.html`
2. **Personalizzare il tema** usando `theme-editor.html` (se necessario)
3. **Configurare** `config.json` per puntare allo scenario e tema corretti
4. **Testare** aprendo `choicemap.html` nel browser
5. **Iterare** fino a raggiungere il risultato desiderato
6. **Pubblicare** copiando solo i file necessari nella destinazione

Versionamento

È buona pratica mantenere versioni degli scenari:

- conservare le versioni precedenti (es. `scenario-v1.json`, `scenario-v2.json`)
- annotare le modifiche significative
- usare un sistema di controllo versione (Git) per progetti complessi

Preparare la pubblicazione

File necessari per l'utente finale

Per pubblicare uno scenario, servono **solo** questi file:

File	Obbligatorio	Note
choicemap.html	Sì	Il Navigator
config.json	Sì	Punta a scenario e tema
defaults.json	Sì	Valori predefiniti
File JSON dello scenario	Sì	Es. scenario-onboarding.json
File JSON del tema	Solo se personalizzato	Es. tema-aziendale.json
Cartella risorse	Solo se usate	PDF, documenti, ecc.

Non includere nella pubblicazione:

- scenario-editor.html (non serve all'utente finale)
- theme-editor.html (non serve all'utente finale)
- shared-styles.css (usato solo dagli editor)
- Scenari in bozza o versioni di test
- File di lavoro dell'amministratore

Configurare config.json

Prima di pubblicare, verificare che config.json punti ai file corretti:

```
{  
  "scenario": "scenario-onboarding.json",  
  "theme": "tema-aziendale.json",  
  "showCredits": false  
}
```

L'opzione showCredits: false rimuove il footer con i crediti, per un aspetto più pulito.

Destinazioni di pubblicazione

Intranet aziendale

Per uso interno, copiare i file in una cartella condivisa sul file server aziendale.

```
\server\condivisione\formazione\  
└── choicemap.html  
└── config.json  
└── defaults.json  
└── scenario-onboarding.json
```

Gli utenti accedono aprendo direttamente il file HTML o tramite un link sulla intranet.

File locali vs web server

Il Navigator funziona anche aperto direttamente come file locale, senza bisogno di un web server, purché tutti i file JSON siano nella stessa cartella.

Sito web pubblico

Per pubblicazione su web, caricare i file su qualsiasi hosting statico:

- **GitHub Pages** (gratuito)
- **Netlify, Vercel** (gratuiti con limitazioni)
- **Amazon S3, Google Cloud Storage**
- Hosting tradizionale

Non serve supporto per PHP, database o altri componenti server-side.

LMS aziendale

Per integrare in un Learning Management System:

- alcuni LMS supportano il caricamento di pacchetti HTML
- altri permettono embedding via iframe
- verificare le policy di sicurezza del LMS specifico

GitHub Pages

GitHub Pages è la soluzione più semplice per pubblicare su web gratuitamente.

Procedura

1. Creare un repository su GitHub
2. Caricare i file necessari (solo quelli per l'utente finale)
3. In Settings → Pages, attivare la pubblicazione dal branch main
4. Attendere il deploy (1-2 minuti)

URL risultante

```
https://nomeutente.github.io/nome-repository/choicemap.html
```

Esempio: se il repository è `training-onboarding` dell'utente `azienda-xyz`:

```
https://azienda-xyz.github.io/training-onboarding/choicemap.html
```

Aggiornamenti

Per aggiornare uno scenario pubblicato:

1. Modificare i file localmente
2. Caricare le nuove versioni su GitHub
3. Attendere il rebuild automatico

Se le modifiche non appaiono subito, forzare il refresh (Ctrl+Shift+R) o attendere qualche minuto per il caching.

Gestire scenari multipli

Un'organizzazione può avere diversi scenari: onboarding, compliance, formazione tecnica.

Opzione 1: cartelle separate

Ogni scenario ha la propria cartella con tutti i file necessari:

```
formazione/
└── onboarding/
    ├── choicemap.html
    ├── config.json
    ├── defaults.json
    └── scenario-onboarding.json
└── compliance/
    ├── choicemap.html
    ├── config.json
    ├── defaults.json
    └── scenario-compliance.json
```

Vantaggi: ogni scenario è indipendente, facile da gestire e aggiornare separatamente.

URL distinti: - `.../formazione/onboarding/choicemap.html` -
`.../formazione/compliance/choicemap.html`

Opzione 2: installazione condivisa

Un'unica installazione del Navigator, scenari multipli selezionabili:

```
formazione/
└── choicemap.html
└── defaults.json
└── config-onboarding.json
└── config-compliance.json
└── scenario-onboarding.json
└── scenario-compliance.json
```

Richiede di rinominare il config appropriato in `config.json` per cambiare scenario, oppure modificare il Navigator per accettare un parametro URL (personalizzazione avanzata).

Embedding in altri siti

Il Navigator può essere incorporato in pagine esistenti tramite iframe.

Codice base

```
<iframe
    src="https://tuodomainio.com/choicemap.html"
    width="100%"
    height="600"
    frameborder="0">
</iframe>
```

Versione responsive

```
<div style="position: relative; width: 100%; height: 100vh;">
  <iframe
    src="https://tuodomainio.com/choicemap.html"
    style="position: absolute; top: 0; left: 0; width: 100%; height: 100%; border:
    none;" 
    frameborder="0"
    allowfullscreen>
  </iframe>
</div>
```

Considerazioni

- Verificare eventuali restrizioni CORS se Navigator e pagina sono su domini diversi
- I link esterni si aprono nell'iframe a meno di `target="_blank"`
- La navigazione da tastiera può comportarsi diversamente dentro un iframe

Sicurezza e accesso

ChoiceMap non ha autenticazione integrata. Chiunque abbia l'URL può accedere.

Per limitare l'accesso:

- Repository privato su GitHub (richiede account Pro per GitHub Pages)
- Hosting su server aziendale dietro VPN
- Autenticazione a livello di web server (basic auth, SSO aziendale)
- URL non indicizzato (sicurezza tramite oscurità, non consigliato per contenuti sensibili)

Per scenari con informazioni riservate, valutare attentamente l'infrastruttura e le policy aziendali.

Riferimento rapido

Checklist operativa per chi conosce già il manuale e vuole un promemoria dei passaggi essenziali.

Avviare il server locale

Windows (doppio clic sugli script batch):

- `start-navigator.bat` → apre il Navigator
- `start-scenario-editor.bat` → apre lo Scenario Editor
- `start-theme-editor.bat` → apre il Theme Editor

Manuale (Mac/Linux o senza script):

```
python -m http.server 8000
```

Indirizzi:

- Navigator: <http://localhost:8000/choicemap.html>
- Scenario Editor: <http://localhost:8000/scenario-editor.html>
- Theme Editor: <http://localhost:8000/theme-editor.html>

Creare uno scenario

1. **New** → crea scenario vuoto
2. **Settings** → titolo, descrizione, autore
3. Selezionare nodo → scrivere **Content**
4. **+ Add Choice** → testo pulsante + nodo destinazione
5. Ripetere per tutti i nodi
6. **Map** → verificare struttura (no bordi rossi = ok)
7. **Save** → scarica il file JSON

Modificare uno scenario esistente

1. **Open** → selezionare file JSON
2. Modificare nodi e scelte

3. Save

Personalizzare il tema

1. Aprire Theme Editor
2. **Brand** → logo, nome azienda, sito web
3. **Typography** → scegliere font
4. **Colors** → sfondo, testo, pulsanti, mappa
5. Verificare nell'anteprima
6. **Save** → scarica il file JSON

Configurare config.json

```
{  
  "scenario": "nome-scenario.json",  
  "theme": "nome-tema.json",  
  "showCredits": false  
}
```

Testare in locale

1. Aggiornare config.json con i nomi file corretti
2. Ricaricare il Navigator (Ctrl+Shift+R)
3. Percorrere tutto lo scenario verificando i percorsi

Pubblicare

File necessari (solo questi):

- choicemap.html
- config.json
- defaults.json
- File JSON dello scenario
- File JSON del tema (se personalizzato)
- Cartella risorse (se usate)

Non includere: scenario-editor.html, theme-editor.html, shared-styles.css

GitHub Pages:

1. Creare repository
2. Caricare i file necessari
3. Settings → Pages → attivare da branch main
4. URL: <https://nomeutente.github.io/nome-repo/choicemap.html>

Codice colori mappa (Scenario Editor)

Colore bordo	Significato
Verde	Nodo iniziale (START)
Arancione	Nodo terminale (END)
Rosso	Nodo orfano (problema!)
Viola	Nodo selezionato

Collegamenti arancioni = loop (tornano indietro).

Markdown nel contenuto

```
# Titolo grande
## Sottotitolo
**grassetto**
*corsivo*
- elemento lista
> citazione
```

Troubleshooting veloce

Problema	Soluzione
Pagina bianca aprendo HTML	Avviare il server locale
Modifiche non visibili	Ctrl+Shift+R per forzare refresh
Porta 8000 occupata	Usare altra porta: <code>python -m http.server 8080</code>
Nodi orfani (bordo rosso)	Collegarli o eliminarli