

# Open Mic Night 2024

11 dicembre

Impact Hub Via Panciatichi, 16, FI

Creare un bot AI in 5 minuti...  
o quasi

Lorenzo Bolognesi

Software engineer @ DXC Technology



**E' possibile creare un  
bot da soli in 5 minuti?**



NÓ



# Langchain



LangChain

Products ▾

Methods ▾

Resources ▾

Docs ▾

Company ▾

Pricing ▾

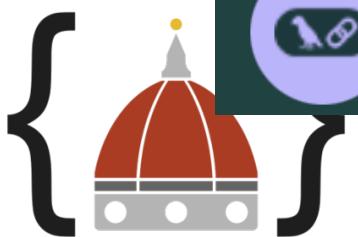
Get a demo

Sign up

# Applications that can reason. Powered by LangChain.

Get a demo

Sign up for free



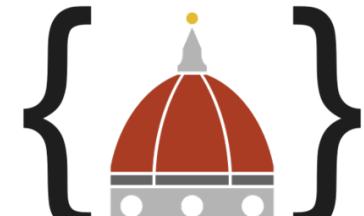
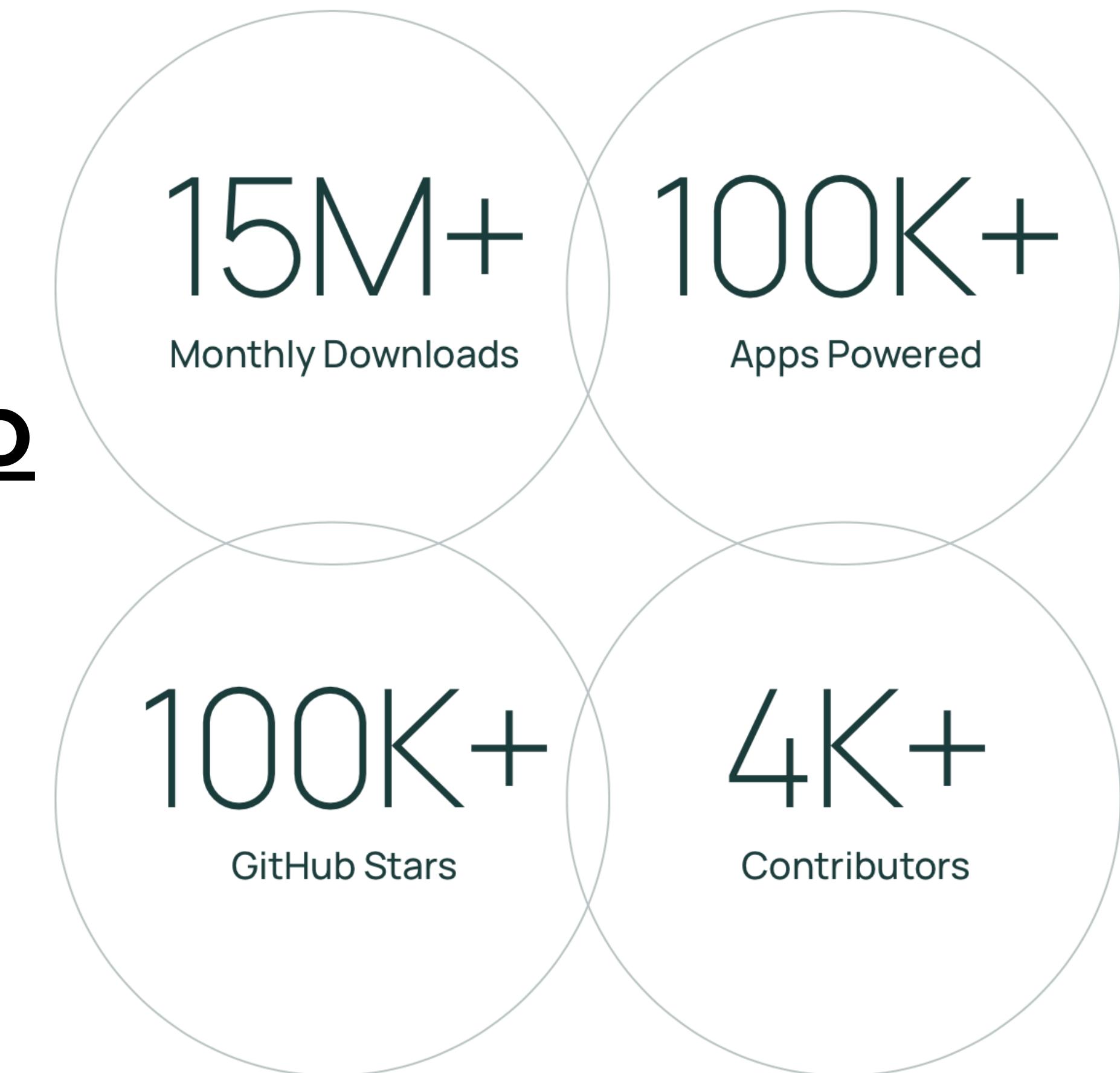
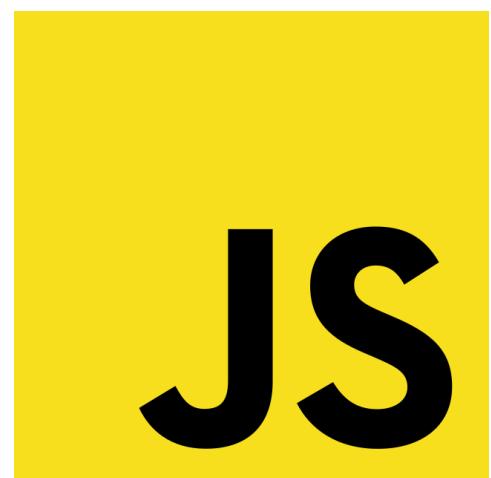
firenze.dev

Da <https://www.langchain.com/>

# Cos'è LangChain?

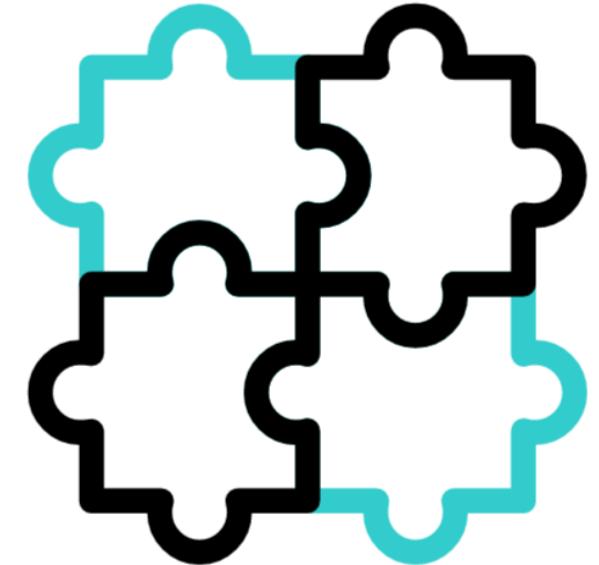
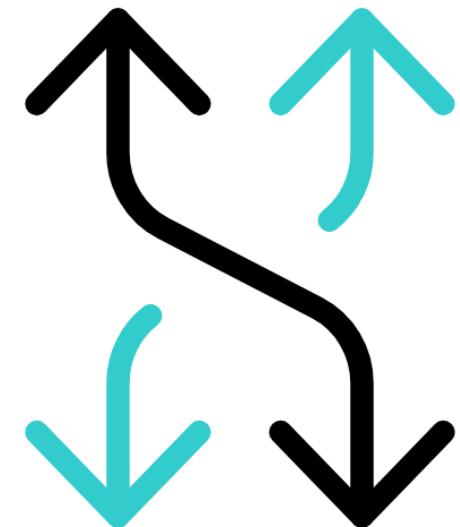
## framework open source

semplifica il processo di  
creazione di applicazioni  
basate su modelli di linguaggio  
di grandi dimensioni (LLM)



# Perché LangChain?

- Semplicità
- Flessibilità
- 600+ integrazioni



Chat History



Today

What can I do with LangChain?

GPT 4o Mini



# What can I do with LangChain?

Done

## Selected Context

Docusaurus | 🐉🔗...

of knowledge or computation. This can include Python REPLs, embeddings, search engines, and more. LangChai...

Architecture | 🐉🔗...

Architecture | 🐉🔗 LangChain [Skip to main content] (#\_docusaurus\_skipToContent \_fallback) # Architecture...

langchain\_nomic 0.1.2 — ...

langchain\_nomic 0.1.2 — 🐉🔗 LangChain 0.2.17 This is a legacy site. Please use the latest v0.2 and v0.3 API...



## Answer

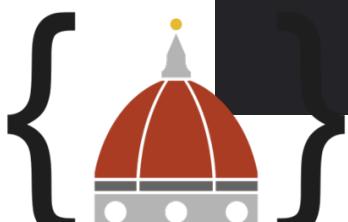
LangChain is a versatile framework designed for developing applications powered by large language models (LLMs). Here are some key functionalities and use cases you can explore with LangChain:

- **Development of Language Model Applications:** LangChain provides building blocks and components to create applications that leverage LLMs effectively. This includes tools for prompt management, model integration, and output parsing.
- **Creating Chatbots:** You can build chatbots that utilize LLMs to generate conversational responses, making them ideal for customer support or interactive applications.

How can I...



Da <https://chat.langchain.com>



firenze.dev

# **Caratteristiche principali di LangChain**



# Interfacce standard

Interfaccia uniforme  
per diversi componenti

Semplifica passaggio  
tra provider differenti



# Memoria

Mantiene lo stato tra diverse chiamate

*Utile per conservazione del contesto*



# Orchestrazione

**Chain:** sequenza strutturata di chiamate a vari componenti

*Collega componenti e modelli in flussi di controllo*



# Integrazioni

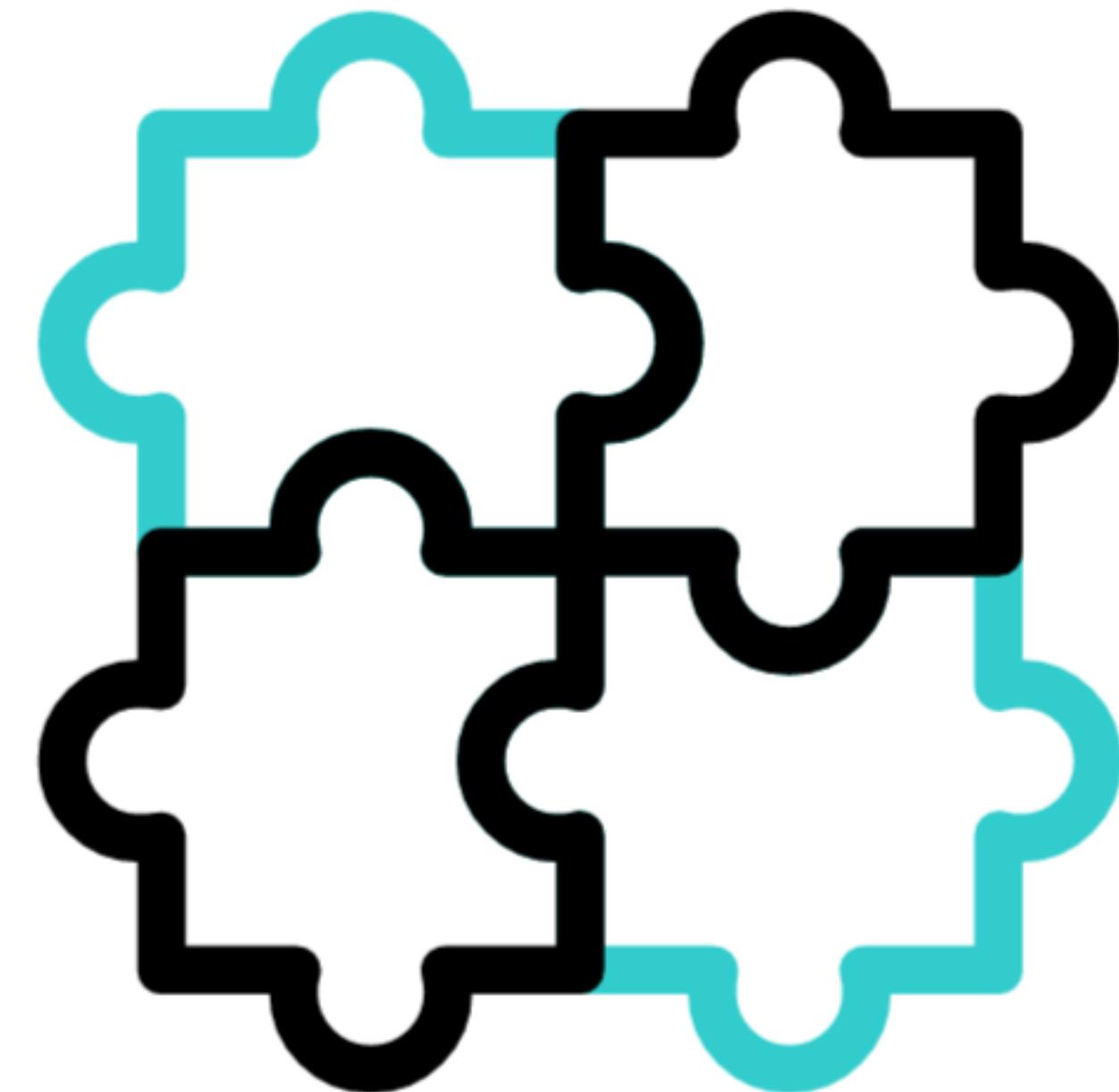
Grande numero di integrazioni di terze parti

[langchain-anthropic](#)

[langchain-ollama](#)

[langchain-openai](#)

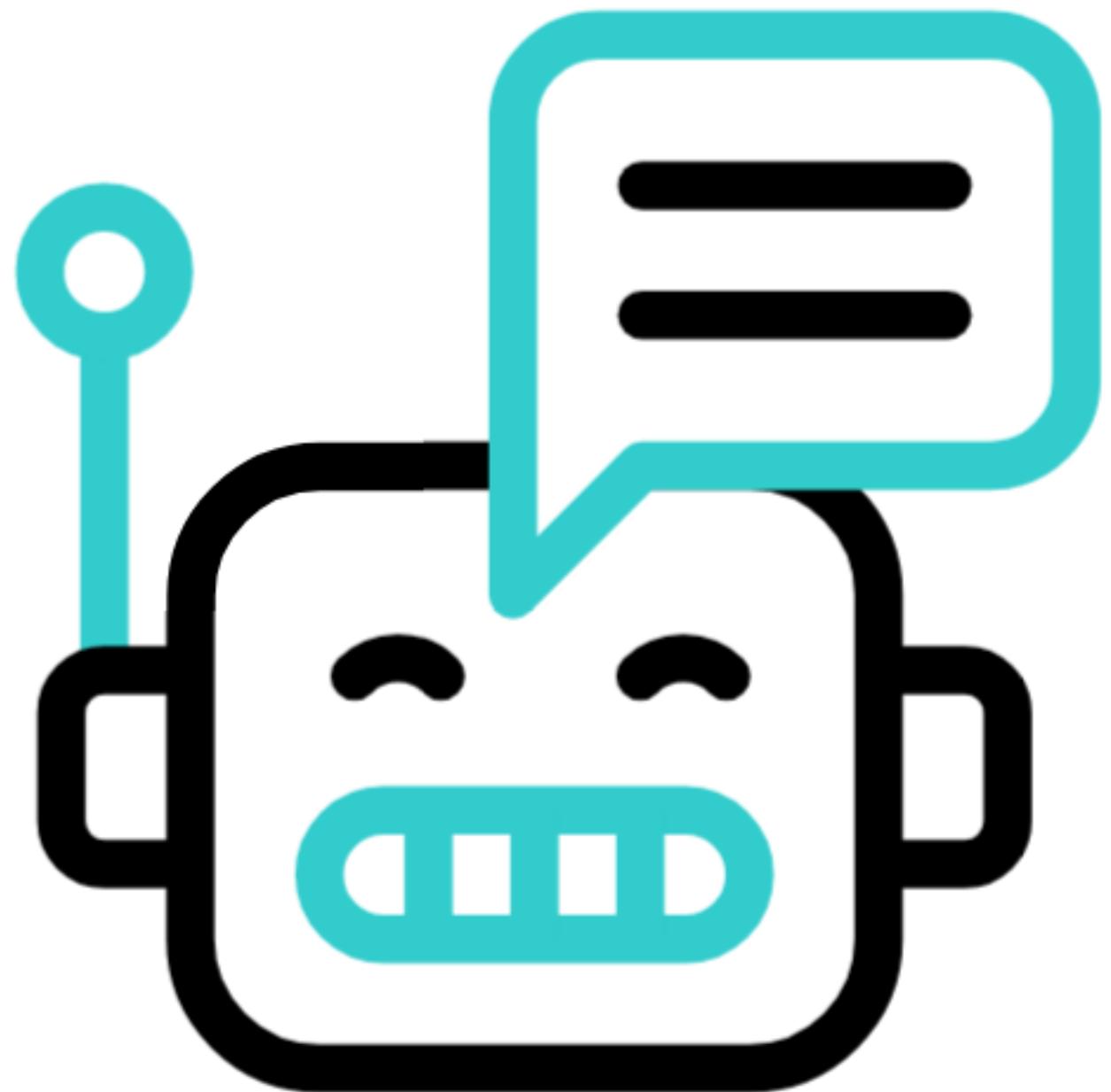
... (vedi tutti qui)



# Agents

**Agent:** utilizza un LLM  
come motore di  
ragionamento

*determina ed esegue una  
sequenza di azioni basata  
su compiti di alto livello*



# Agents

**Tool:** un componente  
**(funzione)** che può essere  
invocato da un LLM

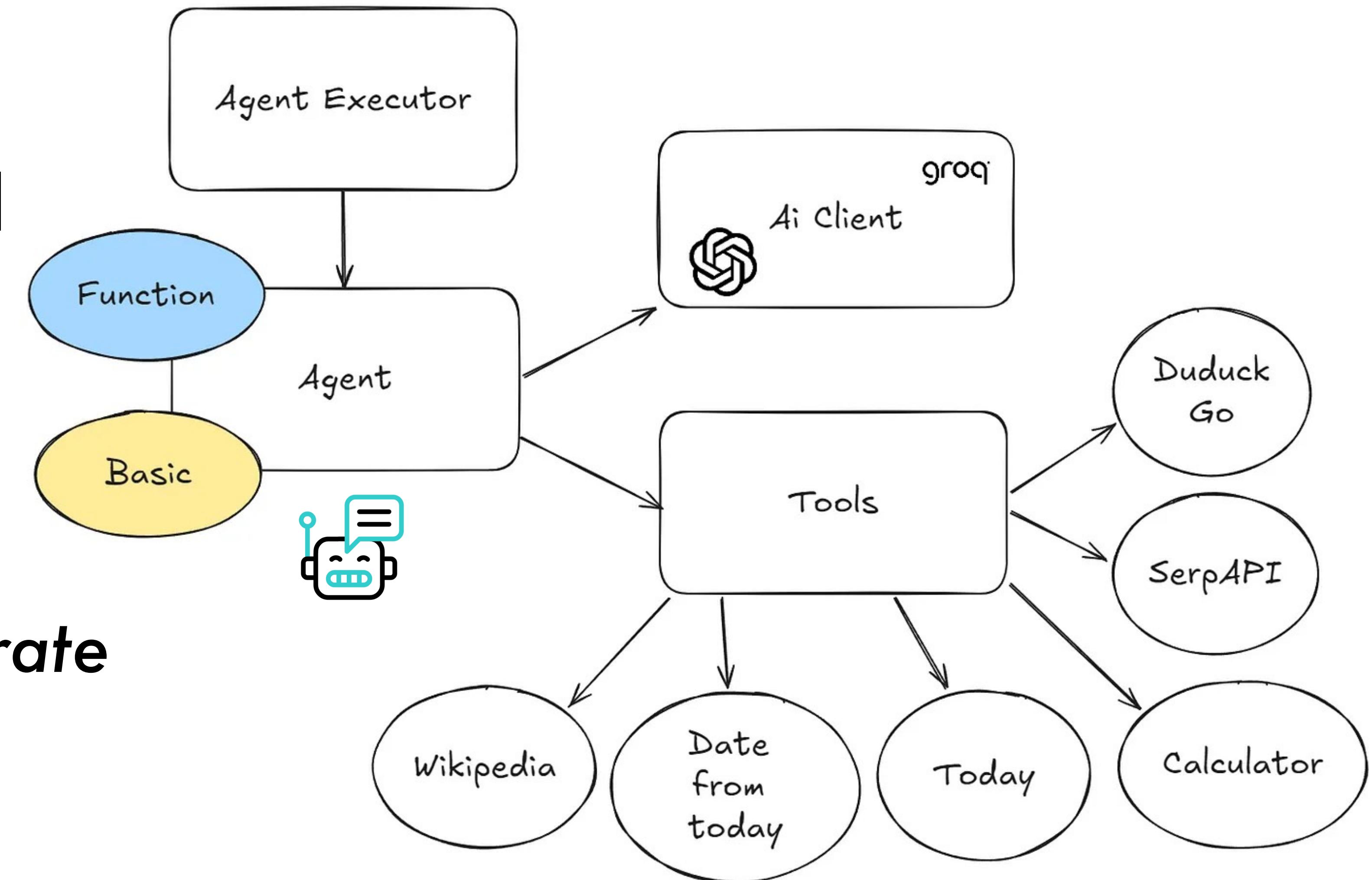
*Es: recupero di documenti,  
chiamate a servizi (API)  
esterni, etc...*



# Agents

**ReAct:** LLM  
interagisce con tool  
esterni

- ricava ulteriori informazioni
- risposte più accurate
- compie azioni



Da <https://medium.com/@gil.fernandes/a-simple-agent-framework-using-react-f2c66e810344>

# Un pò di codice

```
#####
# define prompt
#####
from langchain_core.prompts import ChatPromptTemplate

prompt = ChatPromptTemplate.from_messages(
    [
        (
            "system",
            "Answer the following questions as best you can. Answer in the user's language.",
        ),
        (
            "placeholder", "{messages}",
        ),
    ],
)
```



```
18 #####  
19 # define model  
20 #####  
21 from langchain_openai import ChatOpenAI  
22  
23 model = ChatOpenAI(  
24     model="gpt-4o",  
25     temperature=0,  
26     max_retries=2,  
27     # eventuali altri parametri...  
28 )
```

```
30 #####  
31 # define model (example 2)  
32 #####  
33 from langchain_groq import ChatGroq  
34  
35 model_2 = ChatGroq(  
36     model="mixtral-8x7b-32768",  
37     temperature=0,  
38     max_retries=2,  
39     # altri parametri...  
40 )
```

```
42 #####  
43 # define a simple tool  
44 #####  
45 from datetime import datetime  
46 from langchain_core.tools import tool  
47  
48  
49 @tool new *  
50 def today():  
51     """  
52         Get the current date in the format YYYY-MM-DD.  
53  
54     Returns:  
55         str: The current date in the format YYYY-MM-DD.  
56     """  
57     return datetime.now().strftime("%Y-%m-%d")
```

```
59 #####  
60 # load tools  
61 #####  
62 from langchain_community.agent_toolkits.load_tools import load_tools
```

```
654     """Load tools based on their name.  
655  
656     Tools allow agents to interact with various resources and services like  
657     APIs, databases, file systems, etc.  
  
672     Args:  
673         tool_names: name of tools to load.  
674             llm: An optional language model may be needed to initialize certain tools.  
675                 Defaults to None.  
676             callbacks: Optional callback manager or list of callback handlers.  
677                 If not provided, default global callback manager will be used.  
678             allow_dangerous_tools: Optional flag to allow dangerous tools.  
679                 Tools that contain some level of risk.  
  
688         kwargs: Additional keyword arguments.
```

```
59 #####  
60 # load tools  
63  
64 tools = load_tools(["ddg-search", "read_file"])  
65 tools.append(today)
```

```
67 #####  
68 # create agent  
69 #####  
70 from langgraph.prebuilt import create_react_agent  
71
```

```
210     """Creates a graph that works with a chat model that utilizes tool calling.  
211  
212     Args:  
213         model: The `LangChain` chat model that supports tool calling.  
214         tools: A list of tools, a ToolExecutor, or a ToolNode instance.  
215         state_schema: An optional state schema that defines graph state.  
216             Must have `messages` and `is_last_step` keys.  
217             Defaults to `AgentState` that defines those two keys.  
218         messages_modifier: An optional  
219             messages modifier. This applies to messages BEFORE they are passed into the LLM.  
220  
221         Can take a few different forms:  
222             - SystemMessage: this is added to the beginning of the list of messages.
```

```
67 #####  
68 # create agent  
69 #####  
70  
71 -     agent_executor = create_react_agent(  
72     llm=model,  
73     tools=tools,  
74     state_modifier=prompt,  
75 )
```

```
78 #####  
79 # run agent  
80 #####  
81 user_question = "Che giorno è oggi?"  
82  
83 for s in agent_executor.stream(  
84     input: {  
85         "messages": [("human", user_question)],  
86     },  
87     stream_mode="values",  
88 ):  
89     message = s["messages"][-1]  
90     if isinstance(message, tuple):  
91         print(message)  
92     else:  
93         message.pretty_print()  
94
```

# Esempi

# Proviamo a chiedere



chi sono i fondatori di firenze.dev?

# ...qualcosa non va



- ✓ duckduckgo\_search: fondatori di firenze.dev

^

To find out who the founders of firenze.dev are, I will perform a search to gather the most accurate and up-to-date information.

Action: duckduckgo\_search

Action Input: "fondatori di firenze.dev"



Scopri la Storia e l'Arte dei 7 Santi nella Chiesa di Firenze: Un Viaggio tra Fede e Bellezza Chiesa 7 Santi Firenze: Un Tesoro Nascosto nella Culla del Rinascimento. La Chiesa dei Sette Santi Fondatori a Firenze, comunemente nota come la chiesa 7 santi firenze, è un luogo incantevole che fonde storia, spiritualità e arte. Nasce a Firenze Dorigo e Associati, studio legale tributario con una specifica vocazione per il diritto tributario internazionale, la fiscalità della crisi d'impresa ed il contenzioso tributario, anche in ambito penale. Lo studio comprende sette professionisti ed è stato creato a seguito dello scioglimento un mese fa di DS Tax & Legal, insieme fondato nel 2020 da Dorigo e Giuseppe Salvini - Anch'egli Priore

# ma ci riprova

✓ **duckduckgo\_search:** firenze.dev founders



It seems the search results did not provide relevant information about the founders of firenze.dev. I will refine my search query to get more specific results.

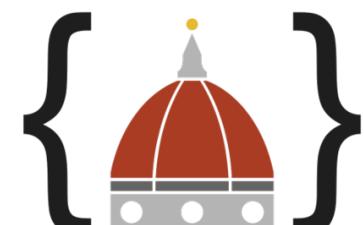
Action: duckduckgo\_search

Action Input: "firenze.dev founders"



firenze.dev presenta l'evento: The unexpected attack: hacking the supply chain - Tanto prima o poi capita a tutti: impara a pensare come un hacker per anticipare gli attacchi. Il 03

Tiratoio, 1, 50124 Firenze FI, Italy firenze.dev. Follow us! Leaders. tiziano pessa. lorenzo spinelli. Topics Programming Innovation. Firenze raised a £750,000 pre-seed funding round to support its launch, with the company now seeking to tap demand for so-called 'Lombard loans' among mass-affluent customers.



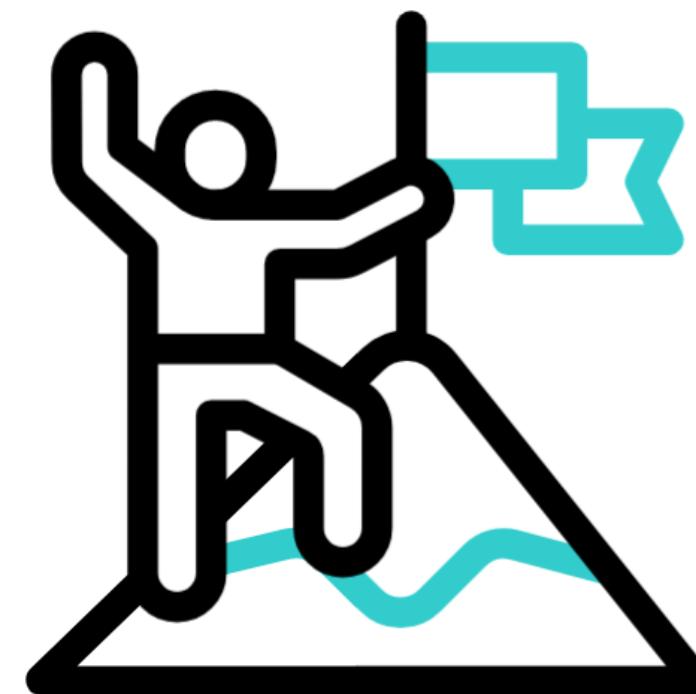
# Soluzione

✓ Complete!

^

I now have the information about the founders of firenze.dev.

Final Answer: The founders of firenze.dev are Tiziano Pessa and Lorenzo Spinelli.



# Demo

# Chatbot

A simple chatbot

Chi ha vinto le ultime elezioni in USA?



# Conclusioni

# Pro

Vanno ridetti?

*Partendo da LangChain ognuno  
può sviluppare le proprie  
integrazioni (magari apprendo pure  
un Pull Request)*



# Contro

Ah! Ce ne sono?

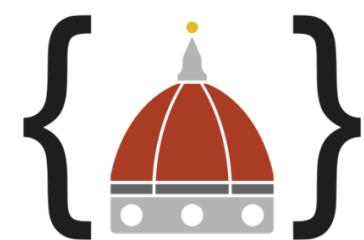
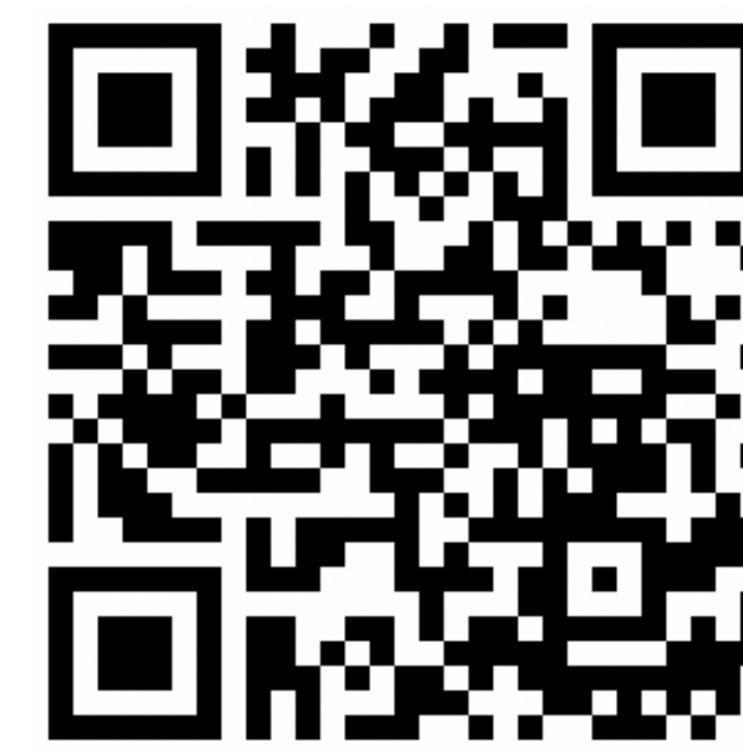
*E' in continuo aggiornamento  
(metodi deprecati, breaking changes)*

*Difficile da testare (ma c'è  
LangGraph)*



# Riferimenti

- LangChain: <https://www.langchain.com>
- LangChain bot: <https://chat.langchain.com>
- LangChain tutorials: <https://python.langchain.com/docs/tutorials>
- Python docs: <https://python.langchain.com/docs/introduction>
- Javascript docs: <https://js.langchain.com/docs/introduction>
- GitHub (python) <https://github.com/langchain-ai/langchain>
- GitHub (javascript) <https://github.com/langchain-ai/langchainjs>
- ReAct prompting: <https://www.promptingguide.ai/techniques/react>
- ReAct article: <https://arxiv.org/abs/2210.03629>



firenze.dev

Icona di Paul J. da <https://it.freepik.com>