OpenAI Agents SDK

What is Agentic AI:

Agentic AI ka matlab aik aisi AI jo sirf order follow nahi karti, balkay khud se kaam karti hai, plan banati hai, soochti hy, aur apne mission ko complete karti phir us mission ko complete karny k liye hurdles aii.

Q- Kya Agentic AI aik saath different tarah ke tasks kar sakti hai — jaise insan?

Short answer:
Haan, agar usko ache se design kiya gaya ho, toh Agentic AI aik insan ki tarah multiple different tasks kar sakti hai — khud soch kar, plan kar ke, aur problems solve kar ke.

Q- What's the difference btw narrow ai, generative ai, and agentic ai?

1. Narrow AI(sir aik kaam kar saky):

Jaise aik robot jise sirf chai banana sikhaya gaya ho.
Jaisy hum kahy k : "Chai banao" → woh sirf chai bana kar dy ga bas.

2. Generative AI:

Mujhe aik love poem likh do,Mujhe aik cat ki funny picture banao, Ya new text/image **generate** kar do.Yani k new cheezy bhi create karta hy.

3. Agentic AI:

Mainy kaha k mujhe aik YouTube channel banana hai — uska naam socho, idea do, video likho, thumbnail banao, aur video post karo."

Agentic AI:

- Sochti hai: "Pehle topic chahiye, phir script, phir thumbnail."
- Khud se tools ka istemal karti hai.
- Ek ek kaam order mein karti hai.

Agentic AI khud se plan bana ke multiple kaam complete karti hai

Q- To phir Agentic AI kya hai? Narrow aur Generative AI to kaam karti hain, to kya wohi Agentic hain?

Nahi, har Narrow ya Generative AI **Agentic nahi hoty**.

Aik AI tabhi Agentic kehlayegi jab:

- Woh **khud soch kar** kaam kare
- Hamey baar baar instructions na deni parein.
- Woh khud planning kare aur step-by-step action le.

What is open ai agents SDK?

SDK = Software Development Kit matlab AI ya app banane k liye ready-made toolbox.

Example:

Socho jese aik lunch box hy jisme sab saman diya hota hai lunch banane ke liye.
Wese hi SDK mein developer ke liye:

- Ready code
- Tools
- Functions
  diye jaate hain, taake woh asaani se kuch create kar sake.

To OpenAI Agents SDK kya hai?

Ek toolbox hai jo developers ko asaani daita hy apna khud ka Agent banane ke liye.Ye SDK humare Liye Kya Karta Hai?

hum sirf bolty hy:

"Hamey aik Agent banana hai jo:

- User se baat kare
- Google Docs open kare
- Email bheje
- Calendar update kare"

SDK hamey deta hai:

✅ Ready-made blocks (tools)
✅ Tarika kaar (flow)
✅ Language understanding (AI brain)
✅ Function calling system
✅ Memory (Agent ko yaad rakhne ki salahiyat)

Q- The Agent class has been defined as a dataclass. Why?

1- What is class:

Ek aisa design ya blueprint jisme hum koi cheez define karty ho or uski properties aur actions.

Example:

```
class Car:
    def __init__(self, color, brand):
        self.color = color
        self.brand = brand
```

2- what is dataclass:

Dataclass Python ki ek shortcut trick hai jo sirf data rakhne wali classes ko banana asan aur clean bana deti hai. For example;

Jab hum sirf values store karna chahte hain (jaise name, age, address), aur inki logic ki zarurat nahi hoti, to tab hum dataclass use karte hain.

Jab hum ek Agent class banaty hy (AI agent ke liye), usme aksar ye cheezein hoti hain:

- Agent ka **name**
- Agent ka **description**
- Uske **tools**
- Uske **skills** ya **functions**

Ye sab cheezein basically sirf **data** hain — unme koi logic ya complex behavior nahi hota.Data class use karny ka faida ye hy hamey jaisy normally jo opar code share kiya hy class ka us tarha nahi define karni parhti simple dataclass ko import kar values define kar daity hy.

Code Example:

```
from dataclasses import dataclass
@dataclass
class Student:
    name: str
    age: int
```

Is se yeh class khud-ba-khud **__init__, __repr__, aur __eq__** jaise methods bana leti hai — bina manually likhe.

Conclusion:

Agent class ko dataclass is liye define kiya gaya hai kyunke iska main purpose sirf data store karna hota hai, jaise agent ka naam, tools, description, aur doosri properties.
Dataclass use karne se code zyada clean, short aur readable ho jata hai, aur Python khud hi kuch useful functions automatically add kar deta hai jaise:

- __init__() (constructor banata hai)
- __repr__() (pretty print karta hai object ko)
- __eq__() (do agents ko compare karne ke liye hota hai)

Yani humy ye sab functions manually likhne ki zarurat nahi padti — Python khud hi bana deta hai.

2a. The system prompt is contained in the Agent class as instructions? Why you can also set it as callable?

Acha ab is question ko phely word by word samjhty hy k system prompt, agent,or callable kiya hy :

What is System Prompt?

Prompt = Prompt ka matlab hota hai: AI ko kuch kehna ya koi kaam dena.Do tarha k prompts hoty hy.

1- USER PROMPT:

Jab user (yaani aap) AI ko koi kaam de:
Mere liye poem likho Ye User Prompt hai.

2- SYSTEM PROMPT:

Jab system AI ko ye batata hai ke tumhara role kya hai, ya tum kaise behave karoge.Ye System Prompt hai.

Example:

AI aik actor hai:

- System Prompt: Director AI ko batata hai: “Tumhara role hai aik helpful assistant banna.”
- User Prompt: Audience AI ko kehti hai: “Ab poem suna do.”

System Prompt work kaisy karta hy hum to simply as a user type kar k khe daity hy lakin ystem prompt kaisy kheta hy?

🔷 **Hidden Instructions:**

Jab hum ChatGPT ya kisi AI tool ko open karty hain,to pehle se ek System Prompt set hota hai.
Jaise:

"You are a helpful, polite, and professional assistant. Answer clearly and concisely."

Yeh AI ko batata hai:

- Tumhara role kya hai.
- Tumhari tone kaisi honai chahiye.

- Tumhein kya karna chahiye ya nahi.

Agar hum apna AI tool bana rahe ho (like in Python ya Node.js), to hum manually System Prompt set kar sakty hy:

```
messages = [
   {"role": "system", "content": "You are a polite AI that explains in Urdu."},
   {"role": "user", "content": "Mujhe AI samjhao."}
]
```

Is tarha sy behind the scene system code k zariye AI ko samjhata hy.

What is Agent?

Agent aik AI worker hy jo hamry liye kaam karta hy jaisy for example aik ai tool hy jo hmaary sy baat kary or hamra kaam bhi kar k dy.

What is Runner?

Runner ka kaam hota hai:

"Agent ko chalana"
Yani hamare banaye huay Agent ko activate karna.

First part : "System prompt Agent class ke andar instructions ke naam se hoti hai"

iska matlab:

Agent class ek AI agent banane ke liye hoti hai.

Uske andar hum yeh bata sakty hy:

- Agent ka naam kya hai?
- Uske paas kya tools hain?

```
Agent(
  name="MathTeacher",
  instructions="Tum aik maths teacher ho jo students ki madad karta hai.",
  tools=[calculatorTool]
)
```

Yahan **instructions** woh message hai jo AI ko batata hai ke uska role kya hai.

Second part: "Why you can also set it as callable?"

Callable ka matlab kya?

Callable = Aisi cheez jo function ki tarah call ho sakti hai.

Yani instructions ke jagah:

- hum seedha ek string de sakty hy. Ya ek function bhi dy sakty hy jo string return kare.

Why use function (callable)?

Agar hum chahty hy ke:

- AI har dafa naye instructions le (situation ke hisaab se),
- Jaise current time, user ka naam, ya weather ke hisaab se,

Toh hum function use karty hy jo har dafa updated instructions dy.

Example: Callable (function prompt)

```
import datetime
def dynamic_prompt():
    current_time = datetime.datetime.now().strftime("%I:%M %p")
    return f"Tum aik smart agent ho. Abhi ka waqt hai {current_time}."

# Jaise yeh agent ban raha hai
agent = Agent(
    name="TimeAgent",
    instructions=dynamic_prompt,  #  pass function
    tools=[]
)
```

2b. But the user prompt is passed as parameter in the run method of Runner and the method is a classmethod?

What is the role of run method?

run() method basically woh jagah hai jahan se AI ka agent kaam karna shuru karta hai.

Runner.run("user ka prompt yahaan aata hai")

To jab bhi hum AI se kuch poochty hy, yeh run() method user ka message accept karta hai aur agent ko bolta hai: **“Chalo! Ab jawab do is input ka.”**

```
class Example:
    @classmethod
    def run(cls, prompt):
        print(f"User ne poocha: {prompt}")

Example.run("Hello!")      #  object nahi banaya, direct chala diya
```

OpenAI Agents mein ye kyu use hoty hai?

- Runner.run() ek **starting point** hai user ke interaction ka.
- Hum run() mein user ka prompt pass karte hain — jo AI agent process karta hai.
- @classmethod use karne ka matlab hai:
    - Hum direct class se run() chala sakte hain.
    - Easy testing, reusability, aur clarity milti hai.

Conclusion:

run() method ek classmethod isliye hai taake hum bina agent ka object banaye seedha us method se user ka prompt bhej saky.
aur har dafa jab user kuch input kary to, woh prompt run() ke parameter mein pass ho jaii, taake agent uska jawab de sake.

3.) What is the purpose of the Runner class?

Runner Class:

Runner class AI Agent ko chalaane, user ke prompt lene, aur uska result deny ki responsible hoti hai.

1. Agent kya karta hai?

Agent ka kaam hota hai:

- Tools ko use karna,Sochna (reasoning), Aur hamary questions ka jawab banana.
-

2. Runner kya karta hai?

Runner class basically AI agent ko "run" karne wali machine hoti hai.
Woh agent ko user ka prompt deti hai aur kehti hai:

- Lo bhai, yeh question hai. Tum jawab do.

Example:

Runner.run("Mujhe aik poem sunao")

Yeh line ka matlab hai:

- Runner user ka prompt ("Mujhe aik poem sunao") leta hai,
- Agent ko deta hai,
- Fir output (jawab) wapas user ko deta hai.

Runner ka kaam detail mein:

Runner class kuch important kaam karti hai:

1. **Prompt lena** (user ka input)
2. **Agent ko chalana** (AI agent ko kaam dena)
3. **Tools activate karna** (agar agent ko kisi external tool ki zarurat ho, jaise calculator)
4. **Result wapas dena** (jo bhi jawab ho, user ko return karna)

Conclusion:

Runner class woh system hai jo AI agent ko chalata hai. Ye user ka prompt leta hai, agent ko deta hai, aur fir result return karta hai.

4.) What are generics in Python? Why we use it for TContext?

Generics:

Generic ka matlab hota hai: aisi cheez jo har tarah ke data ke saath kaam kar sakti ho — bina fix type specify kiye.

Without using generic example:

```
def add(x: int, y: int) -> int:
    return x + y
```

ab yahan par sirf init type kaam kary gi lakin hum chahy k koi specific na hoo hum koi bhi jab dil chahy koi bhi data type use karly to is k generics use karna hogi.

Generics Example:

```
from typing import TypeVar, Generic

T = TypeVar("T")  # Yeh T kisi bhi type ka ho sakta hai

class MyBox(Generic[T]):

    def __init__(self, item: T):

        self.item = item
```

from typing import TypeVar, Generic

typing module se hum TypeVar aur Generic import kar rahe hain.
Ye dono generics ke liye use hote hain — jiska matlab hota hai har tarah ka data accept karna.

T = TypeVar("T") # Yeh T kisi bhi type ka ho sakta hai

T ek type variable hai — iska matlab yeh hai ke T int bhi ho sakta hai, str bhi, list bhi, kuch bhi!

class MyBox(Generic[T]):

Hum ek class bana rahe hain jiska naam hai MyBox,
aur yeh Generic[T] inherit kar rahi hai — matlab:

Yeh class kisi bhi type ka item handle kar sakti hai, jo T ke through aayega.

```
    def __init__(self, item: T):

        self.item = item
```

Jab MyBox ka object banega, to item parameter mein koi bhi data type aa sakta hai.

- self.item = item ka matlab hai:

Us item ko object ke andar store kar do.

```
box1 = MyBox(123)        # int

box2 = MyBox("Hello")     # str

box3 = MyBox([1, 2, 3])   # list
```

Har box mein alag type ka data store ho raha hai — int, string, list — aur yeh possible hai Generic[T] ki wajah se.

Tcontext:

TContext ek generic type variable hai jis ka matlab hy k aik agent ka context (yaani uske paas jo data/memory/tools hain) kisi bhi type ka ho sakta hai.

Kyun use karte hain?

- Har Agent ka context alag ho sakta hai.
- Kisi agent ke paas username ho sakta hai, kisi ke paas tools, kisi ke paas history.

Toh isliye hum generic bana kar kehte hain:

“TContext koi bhi type ki ho sakti hai.”

Real Life Example:

Socho aik school bag hai.
Har student ka bag alag hota hai:

- Kisi ke bag mein books hoti hain,
- Kisi mein laptop,
- Kisi mein lunchbox.

Toh hum bag ko GenericBag kahy gy or usmein TContent kuch bhi ho sakta hai:
book, laptop, pen, etc

Conclusion:

Generic ka matlab hai flexible type — jo kisi bhi data type ke saath kaam kare.
TContext ko generic banane ka matlab hai: agent ka context har project mein alag ho sakta hai, isliye hum usse generic bana ke flexible rakhte hain.