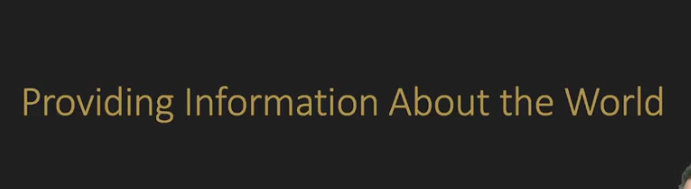
**Providing Agentic AI Information About the World**

****

One of the things that people overestimate when they start working with generative AI is how much generative AI knows because it's so convincing. They often assume knowledge that it doesn't actually have. They assume that it knows all kinds of surrounding information about what they're doing. And also we tend to underestimate how much context is actually needed in order to perform a task. Well, I want you to go back and think about it. When you're building a genetic AI is think about if you tried to get an intern to do something for you. It's day one, they've shown up, they've been at your workplace like one minute, they know nothing about your culture, they know nothing about the layout of your office, they know nothing about your systems in particular. They just have lots of general knowledge about the world, but not about your stuff, not about how you do things, not about your computer systems, not about your goals in life, not about what gas station you stopped at when you were driving to work and what your favorite food is and what you like and don't like, and travel planning. All these things are missing. There's so much information that is fundamentally missing, and it's really critical that we fill that information in whenever we go and get a genetic AI to perform a task.

Now the above content is asking that when we create agent so ask it about our personal info or the info which is very close to ourselves or our country and not available on internet or in its dataset, So it will not provide any info/answer related to it. And if we still want AI to perform our personal task so It is important to provide it our personal info.

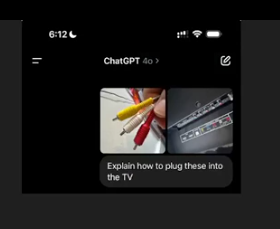
Now let see a example where we provide it the essential info in order to perform a task:



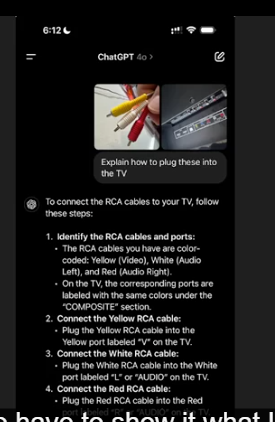
So the other day my son came in and he had found this old Nintendo Wii that we had in the closet, and he said, hey, I want to hook this up to the TV. And we pulled this thing out of the box, and lo and behold, it doesn't have an HDMI hookup. And I thought, what on earth am I going to do? I don't even know if my tv can accept these types of inputs.



And now I go and I try to get to my TV and it's mounted on the wall, and I feel behind it and I feel more than one of those inputs, there's more than three of them. So now I've got a problem because I feel five. There's only three cables and I'm trying to reach behind blindly and figure out what to do. So finally I broke down and I reached out, I took my phone in, I took a picture behind it, and now I've got information about the tv.

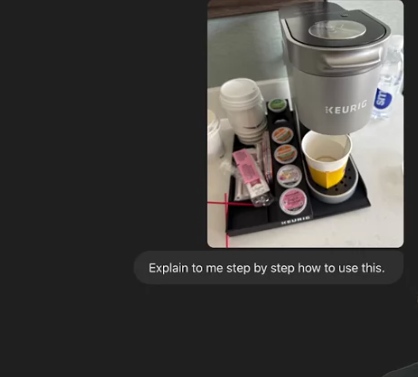


Now, I then decided, okay, let's see if I can get ChatGPT to go and solve this problem with planning. And I said, explain how to plug these into the TV using the ChatGPT app. And then it came back and it said, okay, here's where they go.

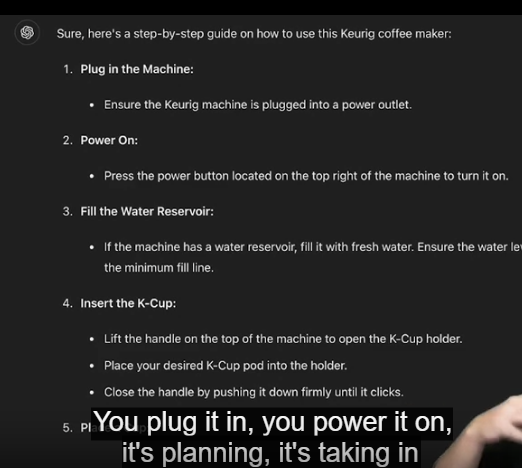


So it built a plan for me to go and connect cable by cable into the tv. Now, the key behind this was that I gave it information about the world. I didn't just say ATV, I didn't say my tv, because my tv is not specific enough  unless it knows my tv and has information, detailed information about it. And even if I said my tv and model, I'd still have to know that it even had information about what the panel looked like and the layout of all of those inputs. There's so much information that's needed to accomplish that task. I also have to show it what I'm working with in terms of wires and the exact colors and how many. And there's so much information that's required in order to perform the task. Now, in this case, I'm giving it really specific information about the world around it. And that's the thing, is, it needs to know the information about the world, the system it's working with, the problem it's solving. It needs a wealth of information to do a good job. Just like that intern on day zero needs information to go and solve the problem.

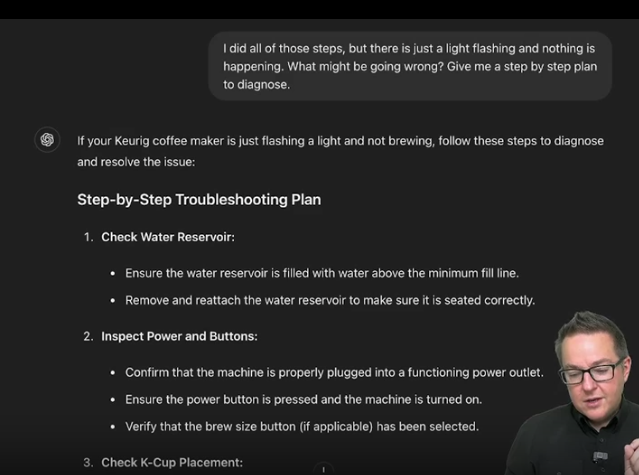
**E.g 2**



Now, let's think about this coffee maker. Explain to me, step by step, how to use this. Now, if I just said a keurig, it may be able to solve it. If I say, explain to me how to use this coffee maker, well, coffee maker is not enough if I don't give it the photo. But the photo is really rich information. One, it says Keurig across the front of it. Two, it's got information about the shape of this thing, what may be the options to use it. And it goes in and breaks it down, and it says, here's what you do.



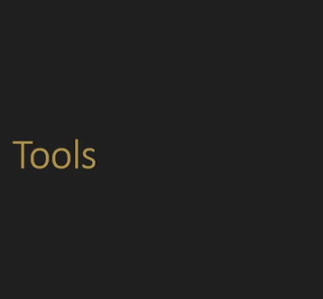
You plug it in, you power it on, it's planning, it's taking in information about the state of the world, then it's deciding what to do. And whenever we begin building these systems, we have to think very carefully about, are we giving the system enough information to perform the task? Does the agent really know all the details and complexities and nuances in order to get to where we want it to go?



And we also have to think about when things don't go right. How do we update its knowledge of the state of the world or something changes? How do we update its knowledge of the state of the world? And all of this has to go into it, because as it's going and executing a plan or telling us to execute a plan, something in the world may change that may force its plan to adapt.  And if we're not feeding it in information continuously to make sure that it's aware of what's happening around it, it will go and make incorrect decisions. So, if you tell it to go create a travel reservation, and then simultaneously, your wife goes and creates a travel reservation independently and you don't update it, it may go and create a duplicate, and that's not what you wanted. And so you have to think about it like that, intern. You have to not only tell it enough information to go and solve the task, but you have to keep it up to date with all the information that it needs. And that means continuously updating it throughout the process so that it knows what it needs to do, but also it needs to know what happened that may affect what it's currently planning to do.

Now above explanation is telling that kay srf asa nhi kay info provide krnay ka bd agent ko independent chor dena blkay usko sth sth update krna hai and stick it to the context.

**Giving Agents Tool**



How do these agentic systems go and interact with the world? Well, we typically have them interact with the world through tools or actions that they can take through those tools.

Now remember we must have to provide the AI the tools or actions so it will generate the response within those tools or action.

**In detail:**

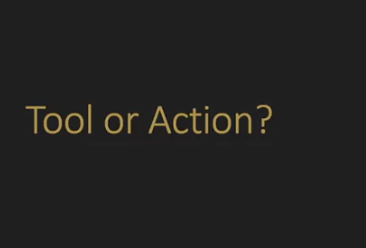
Agentic AI systems interact with the world by using a set of predefined tools or actions to accomplish tasks. The key is to constrain these systems to operate within specific parameters, ensuring they don't perform arbitrary actions. For example, when planning a task like cooking, the AI is given a limited set of tools, such as a skillet and wood fire, and must solve the problem using only those tools. This approach ensures the AI operates within the boundaries of available resources, similar to how it would be restricted when interfacing with rigid computer systems, where it needs explicit actions due to limited capabilities. The concept of tools or actions is context-dependent, with tools being more flexible for human interaction and actions being more precise for computer systems.

Q- Now what does mean by this : similar to how it would be restricted when interfacing with rigid computer systems, where it needs explicit actions due to limited capabilities. The concept of tools or actions is context-dependent, with tools being more flexible for human interaction and actions being more precise for computer systems. ?

Agentic AI systems need to be guided by specific tools or actions to interact with the world effectively. Just like when you give someone a limited set of kitchen tools to cook a meal, these AI systems need to be given clear boundaries on what they can use to solve a problem.

For example, if you're using an AI to plan a trip, you wouldn't want it to suggest flying if you prefer not to fly. So, you tell the AI what options are available—like driving or taking a train. Similarly, when the AI interacts with a computer system, it needs to know exactly what actions it can take, because computers can only do certain things. This makes it important to define clear instructions for the AI, depending on whether it's working with flexible human tasks or rigid computer systems.

Now in simple terms above explanation means that like agar action perform krnay ka liya agay Human hai toh uskay liya be hum agent ko specify krtay hain kay srf yeh tools or resource we have, Toh isi tarah its very much important that agar front pay koi Computer system hai that is going to perform any action which Agent will instruct it, so its important ka Agent ko yeh specify kia jaye kay Computer system kay pass kon kon say tools hain jinkay through wo task perform krega.



Now basically when the performer is Human so will specify the **Tool** to Agent but when performer is a Computer system so we will specify the **Actions** to Agent. Ab basically jo Human hota hai it is more flexible than machine so it can perform many actions using a Tool, But jo computer hota hai uskay pss be bht saray tools hotay hain but wo specific hota hai for performing a specific task. In simple terms wo out of box nhi kaam krskta iss lia hum Ai Agent ko direct wo **Action** btadetay hain jo computer system perform krskta hai So then wo AI agent on the basis of that Actions hi Computer ko task assign krta hai.

**Tool/Actions Description**



When designing agentic AI systems, it's vital to precisely define the tools and actions available to the AI, ensuring it can perform tasks correctly within a specific context. Here's a detailed summary, along with explanations of the examples provided:

**Summary:**

Agentic AI systems interact with the world by using predefined tools and actions, which must be clearly defined to prevent misunderstandings. While the AI can easily understand common tools like pots and skillets, custom tools in computer systems—often with unique names and functions—require detailed explanations. The AI relies heavily on these descriptions to perform tasks accurately. Poorly named or inadequately described tools can lead to mistakes, where the AI may use tools incorrectly. Therefore, it's crucial to give the AI clear, descriptive names and detailed information about each tool's function.

**Example 1: \*\*Cooking with Limited Tools\*\***

- \*\*Scenario\*\*: Imagine you're using an AI to help cook a meal, but you only provide it with a few tools: a skillet, a one-quart sauté pan, and a wood fire. The AI knows it can only use these tools and must guide you step-by-step to cook a dish.

- \*\*Explanation\*\*: The AI doesn't have access to every kitchen gadget; it must solve the problem using only what’s available. If you say you want to make pizza on the wood fire, the AI will guide you through each step—preparing the dough, heating the skillet, and cooking the pizza—using only the tools you specified. This example shows how AI can solve problems within a set of constraints, much like it would in a real-world scenario with limited resources.

**Example 2: \*\*Escaping an Alien Spaceship\*\***

- \*\*Scenario\*\*: In a more imaginative scenario, suppose you're trapped in an alien spaceship and need the AI's help to escape. You give the AI a set of alien tools with strange names: "X155" to prepare alien pizza, "Q63" to open a dimensional portal, and "L199" to play Beatles music on a loop.

- \*\*Explanation\*\*: The AI doesn’t know what these tools do just by their names, so you must describe each tool’s function. For example, "X155" prepares alien pizza, which could distract the aliens, allowing you to escape. "Q63" opens a portal to another location, and "L199" plays distracting music. The AI uses these descriptions to come up with a plan: first, prepare the pizza to distract the aliens, then open a portal to escape, and finally, play music to cover your tracks. This illustrates how crucial clear tool descriptions are for AI to make the right decisions in unfamiliar situations.

**Example 3: \*\*Tool Naming Importance\*\* (very important step)**

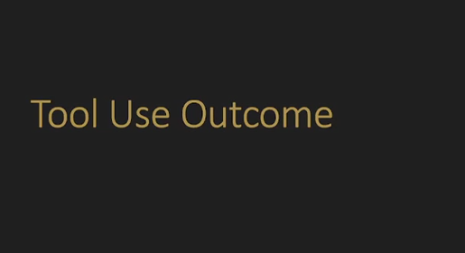
- \*\*Scenario\*\*: After describing the alien tools, you rename them to more intuitive names: "makeAlienPizza," "openDimensionalPortal," and "playBeatlesMusic." You then ask the AI to help you escape again.

- \*\*Explanation\*\*: With these clearer names, the AI can easily understand and suggest the right steps—like making pizza to distract the aliens—because the tool names directly describe their functions. However, if you change the names to something ambiguous, like "mkpz" or "odprtl," and remove the descriptions, the AI might get confused. For instance, it could mistakenly use "mkpz" to create a map instead of making pizza because it doesn't understand the tool’s purpose anymore. This example underscores the importance of both descriptive naming and detailed explanations in ensuring the AI correctly interprets and uses the tools.

**Conclusion:**

So in simple terms , jab be Ai ko koi tool or action specify krna hai toh agar toh wo name say hi smjh aa raha hai toh its ok otherwise usko proper decscribe krna hai like if we talk in term of computer so lets suppose humnay koi asi device create ki jo itni popular be nhi hai or AI ko uska pta be nhi hai and uska name hai **RTZ** ab wo apnay name say be specify nhi ho rhi kay actual what is the purpose of this so at that time it is crucial to tell the purpose of that device to AI.

**Tools feedback/outcome**

****

When building agentic AI systems, one of the critical aspects is guiding the AI through a sequence of actions by using predefined tools. These systems need to be provided with explicit feedback on the outcomes of the tools they use since they cannot directly observe the results. This feedback loop is crucial for the AI to adapt and make informed decisions on subsequent steps.

**Example 1: Cooking with a Microwave**

- \*\*Scenario\*\*: Suppose you instruct the AI to help reheat a quesadilla in a microwave. The AI is given a set of "tools" (actions) it can use, such as "microwave increase time," "microwave start," "microwave open door," and so on.

- \*\*Process\*\*:

- The AI first instructs you to insert the quesadilla into the microwave using the "insert food in microwave" tool.

- After you perform the action, you must provide feedback to the AI: "Result: Food in microwave."

- The AI then tells you to increase the time, and you might respond, "Result: Time increased by 5 seconds."

- The AI adapts, realizing it needs to continue increasing the time to reach the desired 60 seconds.

- If an error occurs, like the microwave door being open, you inform the AI: "Error: Door open."

- The AI responds with the correct action, "Use the 'microwave close door' tool."

- This back-and-forth continues until the quesadilla is successfully reheated.

**Key Concepts:**

1. \*\*Tool Feedback\*\*: The AI needs explicit feedback on the outcome of each tool or action it instructs. This feedback allows the AI to adapt and decide the next action based on the current state.

2. \*\*Error Handling\*\*: Error messages must be clear and meaningful. If the AI receives an ambiguous or incomprehensible error code, it may not know how to proceed. Providing a clear explanation, like "Door open" instead of an error code like "Error 32," ensures the AI understands the issue and can suggest the appropriate corrective action.

3. \*\*Simulated Feedback Loop\*\*: While this example involves manual feedback, the goal is to have the actual computer system provide real-time feedback to the AI. For instance, the microwave could directly inform the AI that the door is open or that the time has been set, eliminating the need for human input in this loop.

\*\*Conclusion\*\*:

Now basically while we planning to perform in **step by step** order , So it is very important to provide the AI Agent the outcome of the action performed , It is because the AI has told you that how to perform that task using the tools you specify to it But it does not know the outcome of that tools so its our duty to provide it the Tools outcome so that it can give the next instruction on the basis of previous outcome by Tool. And the second important thing is if any error occurs so its important to properly provide the error description to AI soo it can provide you with the solution But if you only provide the error code like 33 so it cant understands it and refuses to answer you.

**In-Context Learning and Agents**

In the development of agentic AI systems, providing explicit instructions is one approach to guide the AI in solving tasks step by step. However, another powerful method is in-context learning, where the AI learns by observing examples of problems being solved correctly. This method is similar to how we might teach an intern by showing them examples of completed tasks. By giving the AI examples, it can infer the correct use of tools or actions, even when not explicitly told what each tool does. In-context learning is crucial for improving the performance and predictability of agentic AI systems, especially when interacting with complex or rigid computer systems.

**\*\*Example 1: Escaping an Alien Spaceship\*\***

- \*\*Scenario\*\*: In this scenario, the AI is given a set of tools with meaningless names like "X155," "Q63," and "L199," and no descriptions are provided. The AI initially struggles to use these tools correctly because it doesn’t know what they do.

- \*\*Process\*\*:

- Instead of explicitly telling the AI what each tool does, the developer provides examples of how these tools can be used to solve problems.

- For instance, the AI is shown an example where the problem is "feeling hungry," and the tool "Q63" is used to prepare "alien pizza." From this, the AI learns that "Q63" is related to food preparation.

- Another example shows the tool "X155" being used to open a wormhole to Vanderbilt University when the problem is needing to move to another world.

- Through these examples, the AI learns to associate the tools with specific outcomes and can apply this knowledge to new problems, like escaping the spaceship.

**\*\*Example 2: Reheating Food in a Microwave\*\***

- \*\*Scenario\*\*: The AI is tasked with reheating food in a microwave, and it must learn how to control the microwave’s time settings through examples.

- \*\*Process\*\*:

- The developer provides the AI with a sequence of actions to follow, such as "microwave increase time" to reheat food for a specific duration.

- In one example, the problem is reheating leftover pizza for 20 seconds, and the AI is shown how to achieve this by increasing the time in 5-second increments four times.

- Another example involves softening ice cream, requiring the AI to increase the time twice (10 seconds total).

- With these examples, the AI learns the pattern of using the "microwave increase time" tool multiple times to reach the desired total time.

- When asked to melt shredded cheddar on nachos, the AI successfully follows the learned pattern, opening the microwave door, placing the food inside, and then applying the correct number of time increments to melt the cheese.

**\*\*Key Concepts\*\*:**

1. \*\*In-Context Learning\*\*: This method allows the AI to learn by observing examples rather than relying solely on explicit instructions. By seeing how problems are solved with certain tools, the AI can infer the correct use of those tools in new situations.

2. \*\*Pattern Recognition\*\*: In-context learning helps the AI recognize and follow patterns in problem-solving. Once the AI learns a pattern from examples, it can apply this pattern to similar problems, making it more efficient and adaptable.

3. \*\*Application to Rigid Systems\*\*: In-context learning is especially valuable when interfacing with rigid computer systems that require specific, structured inputs. By teaching the AI through examples, it can learn the acceptable patterns and formats needed to interact with these systems without causing errors.

\*\*Conclusion\*\*: