The same of the sa
LANGCHAIN is a rimplified
framowork that eimplifies the process
or building LIM powered Apps
Langchaun has
-) tools 7 1
-7 Abstration)
-> Building am LLM App isn't straightforwird
Lost
Data
V — Hustonia
User -> LLM - History
Snows
APICOUS
Other Google LLM Search
-> To build an intiligent Applike above, you will
need Lang Chain.
-> Long Chain has name interface across
all LLM vendors.
> Long Chaun (LC) have prompt management
, optimation, residuation
7 LC also par document to aders (. pdf,
word, emails, etc)
-> LC supports tool calling, etc
7) C . L . L . L . L . L . L . L . L . L .
-> LC creates abstractor
CACATIS -
-> Lang Chain API
-> Lang Charin API
-> Lang Chain API -> Lang Chain
-> Lang Chain API -> Lang Chain
-> Lang Chain API Long Chain Long Chain Long Chain
-> Lang Chain API Long Chain Long Chain Long Chain
-> Lang Chain API Lang Chain
-> Lang Chain API Lang Chain
-> Lang Chain API Lang Chain
Lang Chain API Lang Chain Lang Chain Lang Chain Lang Chain Model Io
Larg Chain Larg Chain Larg Chain Larg Chain (ommity) Larg Chain Gore Larg Chain Model Io Emprenon Legeuge Document Load
Large Chain Large Chain Large Chain Large Chain Large Chain Core Large Chain Emprenar Largeuge Nounet Lord Vector Store Agent too by
Larg Chain API Larg Chain Larg Chain Larg Chain Core Lag Chain Emprena Legage Document Load Vector Store
Lang Chain All Long Chain Long Chain Long Chain Long Chain Core Long Chain Core Long Chain Core Model Io Exprenon Lapunge Down-Lond Vector Store Agent too by
Lang Chain All Long Chain Long Chain Long Chain Long Chain Core Long Chain Core Long Chain Core Model Io Exprenon Lapunge Down-Lond Vector Store Agent too by
Large Chain Large Chain Large Chain Large Chain Large Chain Core Large Chain Emprenar Largeuge Nounet Lord Vector Store Agent too by
Lang Chain All Long Chain Long Chain Long Chain Long Chain Core Long Chain Core Long Chain Core Model Io Exprenon Lapunge Down-Lond Vector Store Agent too by
Large Chain Af I Large Chain Large Chain Large Chain Large Chain Large Chain Large Chain Model Io Emprenon Lagrage Document Look Vector Store Agart too by Contains a lot of prompts Contributed by Community.
Large Chain API Large Chain Large Chain Large Chain Large Chain Large Chain Longe Chain Model Io Emprenon Legange Document Lond Vector Store Agarthoohy Tonge Chain Hub Conteains a lot of prompts Contributed by Community.
Large Chain API Large Chain (ommonty) Large Chain (ore) Large Chain
Lang Chain Af I Lang Chain Lang Chain Lang Chain Lang Chain Lang Chain Lang Chain Model Io Emprenon Legeuge Down-tlood Vector Store Agent too by Contains a lot of prompts Contributed by Community. Python 3.11.9 Cagchain Pipeny Lang Chain openai
Large Chain API Large Chain Large Chain Large Chain Large Chain Large Chain Large Chain Model Io Emprenon Largeuge Document Lord Vector Store Agent too by Contains a lot of prompts Contributed by Community. Python 3.11.9 (Lagdown
Lang Chain Af I Lang Chain Lang Chain Lang Chain Lang Chain Lang Chain Lang Chain Model Io Emprenon Legeuge Down-tlood Vector Store Agent too by Contains a lot of prompts Contributed by Community. Python 3.11.9 Cagchain Pipeny Lang Chain openai
Large Chain API Large Chain Large Chain Large Chain Corre Large Chain Corre Large Chain Gore Large Chain Gore Large Chain Gore Model Io Emprenon Legeuge Downstrond Vector Store Agathory Agathory Contains a lot of prompts contributed by Community. Python 3.11.9 Largehain - openai Largehain - community
Lang Chain API Lang Chain Lang Chain Lang Chain Lang Chain (orner Language Document Lond Vector Store Agent too by Contains a lot of prompts Contributed by Community Python 3.11.9 Lang Chain - openai Institute (any chain - community hay chain - community
Large Chain API Large Chain Longe Chain Longe Chain Longe Chain Longe Chain Longe Chain Longe Chain Model To Emprenon Legrage Neutor Store Aget too by -> Longe Chain Hub Contains a lot of prompts Contributed by Community. Python 3:11.9 Lagdown Pipenul Lagdown Lagdown Pipenul Lagdown Lagdown Lagdown - commity hagdeinhub black
Large Chain API Large Chain Large Chain Large Chain Large Chain Large Chain Large Chain Longe Chain Model Io Emprenon Legrage Neutor Store Aget too by -> Large Chain Hub Vector Store Aget too by Contains a lot of prompts contributed by Community. Python 3:11.9 Largehain Property Largehain-commity haydrainhub black

Also install pethon-dotern

Fromp Template:
Reompt is input to UM (instructions
tdet2)
Krompt Template helps to translates wer inputs, parameters into instruction
for LLM
Dictionary (Key: Value)
Prompt Template
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
Prompt Volue
] ilb to TTW: dronged bround
LLM
Chat Models:
Pronce aress to LLM via
Chatmodels
Listof Mensages (1P)
Chartmadal
J
Musage OIP
> They also support:
→ tool calling
-7 Structored OFP
-> Mutterned al ipp of P.
-> By Combiny prompt templates,
chatmodely, we can passitate to
_ _ /
-7 We can combine this with external APIs;
-7 We can combine this with external APIS; pdf readers, the writers, etc
-7 We can combine this with external APIs;
-7 We can combine this with external APIs; pdf readers, the writers, setc in Long Chain to properly make an Applicate -> We can combine / Chain all there together
-7 We can combine this with external APIs; pdf readers, the writers, setc in Long Chain to properly make an Applicate
-> We can combine this with external APIs; pdf readers, the writers, etc in Long Chain to properly make an Application -> We can combine / Chain all there together to make an Application
-> We can combine this mth external APIs; pdf readers, the writers, setc in Long Chain to properly make an Applicate -> We can combine / Chain all there together to make an Application -> Agent: In LLM world, an agent
-> We can combine this with external APIs; pdf readers, the writers, etc in Long Chain to properly make an Application -> We can combine / Chain all there together to make an Application
→ We can combine this mth external APIs; pdf readers, the writers, etc in Long Chain to properly make an Applicate The control of their all there togethered to make an Application Agent: In Lim world, an agent is a loop that repetitedly reads a conversation, deades what actor or tool all to make, feeds the resulting
→ We can combine this mth external APIs; pdf readers, the writers, setc in Long Chair to properly make an Applicate The consine / Chair all there together to make an Application Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal is
→ We can combine this mth external APIs; pdf readers, the writers, etc in Long Chain to properly make an Applicate The control of their all there togethered to make an Application Agent: In Lim world, an agent is a loop that repetitedly reads a conversation, deades what actor or tool all to make, feeds the resulting
-> We can combine this not believed APIs; pdf readers, the writers, setc in Long Chain to properly make an Application -> We can combine / Chain all there together to make an Application -> Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal is met
→ We can combine this mth external APIs; pdf readers, the writers, setc in Long Chair to properly make an Applicate The consine / Chair all there together to make an Application Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal is
-7 We can combine this with external APIs; pdf readers, the writers, etc. in Long Chain to properly make an Application. -> We can combine / Chain all there regetled to make an Application. -> Agent: In Lim world, an agent is a loop that representedly reads a conversation, deades what actor or tool all to make, feeds the result back into the modal until the good; met -> Long Chain is a Python framework that gives us buildry hlocks. (prompt templates, Chains, marrories,
→ We can combine this with external APIs; pdf readers, the writers, setc in Long Chain to properly make an Application We can combine / Chain all their together to make an application → Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the good is met > Lang Chain is a Python framework that gives us buildry blocks
→ We can combine this with external APIs; pdf readers, the writers, setc in Long Chair to properly make an Application → We can combine / Chairn all their together to make an Application → Agent: In Lim world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goalis met → Lam g Chairn is a python framaewor that gives us buildry blocks (prompt templates, Chairs, marrories, callbacks, tool wrappers) to brild LIM appo
-> We can combine this mth external APIs; pdf readers, the writers, etc in Long (hown to properly make an Applicate -> We can combine / Cheirn all their together to make an Application -> Agent: In LLM world, an agent is a loop that represently reads a conversation, decides what actor or tool all to make, feeds the result back into the model until the good is met -> Lang (hown is a pythen framewow that gives us buildry blocks (prompt templates, chains, mamories, allbacks, tool wrappens) to brild LLM appo
-> We can combine this mith external APIs; plf readers, the writers, etc in Long Chain to properly make an Applicate -> We can combine / Chein all their together to make an application -> Agent: In Lim world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal is met -> Lang Chain is a Python framewow that gives us buildry blocks (prompt templates, chains, mamories, allbacks, tool wrappend to brild LIM appo -> REA (T -> It is a prompting pattern where the model (LLM) emits an interleaved / Hought ->
-> We can combine this with external APIs; plf readers, the writers, etc in Long Chain to properly make an Applicate -> We can combine / Chein all their together to make an Application -> Agent: In LLM world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal is met -> Lang Chain is a python framework that gives us buildry blocks (prompt templates, chains, mamories, allbacks, tool wrappend to brild LLM apple -> REA (T -> It is a prompting pattern where the model (LLM)
-> We can combine this noth external APIs; pdf readers, the writers, extc in Long Chain to properly make an Application >> We can combine / Chain all there together to make an application >> Agent: In LLM world, an agent is a loop trad representedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal; met >> Lang Chain is a python framework that gives us buildry blocks (prompt templates, Chains, mamories, allbooks, tool wrappend to build LLM apple -> REA (T -> It is a prompting pattern where the model (LLM) emits an interleaved thought -> Action > Observation trace
-> We can combine this mith external APIs; plf readers, the writers, etc in Long Chain to properly make an Applicate -> We can combine / Chein all their together to make an application -> Agent: In Lim world, an agent is a loop that repeatedly reads a conversation, deades what actor or tool all to make, feeds the result back into the model until the goal is met -> Lang Chain is a Python framewow that gives us buildry blocks (prompt templates, chains, mamories, allbacks, tool wrappend to brild LIM appo -> REA (T -> It is a prompting pattern where the model (LLM) emits an interleaved / Hought ->

Tark:
-> We want to build a popular person
information sommerizer.
-> This is how you would doit wrg
Long Cherin madules.
information = · · · ·
some intornation about en
Popular pernon
Summary_template=
given the information: { information}
about a porson, I want you create:
1. A Charat Cummary
2- two intersing towns
2 100 (11/20213 19003
sommary - prompt - template - Prompt Template
[nput_vanables = ['intermation']
templete = summery_templete)
(Im = (hat Open A] (temperateure = 0,
model_nome = gpt-3.5-turi
Chain = Summary - prompt-template DUm
piperymbo
725 = Chevin. invoke (input = { information:
internata)
prot(res)

Umny open Source Models with Lay chain
- Langthoun con be configured to be
used with Ollama, LLAMA 3
Mixtral, etc (local models)
- However the latency will soffer as
the models take forever to von.
-> pipervinstall langchain-ollama
(hat Open AI) Object instantation
(hat Open AI) Object instantation
uva
[chatolloma].
eg: [um = chatolleuna (model= 'llamna3'
Most of the heavy lifting comes from this line
this bre
chain = summary_prompt_template / Mm
StroutputPerner
have we are une operator overloading
have we are use operator ourbody to chain prompt all LLM7 format OIP
TOUP
Strontput
Parrer
>The ice braker application
-> getters information on
agnen individual (text tex, -> Sizmmarizes it anked in)
-> Summarizes it unkerty
> display to was
L> Building this App usy:
) Chains
2) Agents
3) Custom Agents
1) That & that k, th
5) Output Parsers

Scrupin 10 Signup
Serne ATI
De-Artice In-A
Scrope Linkod TN Profile and
Scrope Linked In profile and
der
-> Cot data u langan malla
as JSON.
> Pous to LLM if recovery make decisions/generate of P based on that profle
mare decisions / general
of parea on the prore
dela.
LLM isn't connected to the Web.
They can be Convidered as a repo
of state information
> Agent/bot can connect to both
internet through tools.
-> tooks are scripts or 3rd pointy
Sernles
-> LLM with a prompt can breakdown
Emospard representations of in
achere those tasks (use tool
Cerlls).
-> These autions can be then thoused
to one another.
-> Agents Think Sime uputh stops
Think ->do taxt.
> de tare D
Prompt ->cho formec
hupme LANGCHAIN :
AGENT AGENT
Unes Execute
Unes Exempt
Uner Exemple oxam Idid what
Unes Exempt
Unes Exempt orange (did what were asked me to
Oray Idid what were asked me to -> Chain of Thought is a prompt
Unes trends Okayo Idid what were asked me to

some what wer is asking the
agent to rolve
The idea of Agus is to me larguage model to choose a repuelled tents to
fake.
-> Agents use LLM as a reasoning engine to determe which action
40 to toke.
-> Build a rearch agent.
La hadrance from text box
L) rearch name (REACT agest)
Set linned in profile bound on UKC.
We builden rooth too!
Lo unkedin_lookup_tool.py
-> Import (prompt template
trom (to to a
from create real agent
large chain Agent executor hub 7 Premade
large chain Agent executor hub 7 Premade promp ts.
lang chain Agent executor hub 2 premade prompts.
large chain Agent executor hub premade prompts.
lang chain Agent executor hub premade prompts. template 1 (promptemplate obj) Tum obj
large chain Agent executor hub premade prompts.
lang chain Agent executor hub hub premade prompts. template 2 (promptemplate obj) tum obj too's for agent obj
lang chain Age t executor hub hub premade prompts. template obj tum obj thoo's for aget obj
lang chain hos hub premade prompts. template obj Lim obj thoo's for eget obj This obj and create
lang chain Agent executor hub hub cutour hub Premade prompts. template 7 promptemplate obj Lim obj this obj mill Agent This obj mill
lang chain hos hub premade prompts. template obj Lim obj thoo's for eget obj This obj and create

> Main idea behind icebraker App
-> Lookup linkedin wer name based
DN user densiption (reter URI
-> Pull Linked in profile and Summarre
Content for ice breaker
Steps:
1 L> Search API -> Listy wery
in Jon
2 Lo REALT -> LLM-75; Hers the hat of SO PROMPT URL
3 L> Scrape > Hnked In URL > JSON API
API
4 L>Sommarize Tson was a prompt
V
->Briefintro
-> 2 interestry
-> Dutpt Parsers:
- LLM anally OJP text
- Ortput Porsers take tout
and transform into 350m,
csv, etc
-> Py Dentic is a per lage that helps
in schema and data validation.
-> Tobreaka Off -> Tommang-istr
2 Facts > [str]
V
If LLM OJP a Stong with
JSON out est
J 3801 - Q 64 -
the Debut Rarrer is a Dire
the Output Parker is going to wrapit in a Custom
rent object > Summare.
> Summary.
Tes-summary -> Str

res-facts -> List(str)

-> Creating a full Stack App:
Using FLASK
-> WebPage
-> IP User rome
-> Run button
_
Facts) Web Pines
-> hines Sommany Shows upon Facts Web Pringe IcreBreakord
-> I cobraker return Summary abject.
rather than Str.
-> index.html
Lat box
button dick action
L) Rethon function I call
ice by earler
-> Response is rendered to UI
-> Long. Smith: Service by Leng Chair
offers Observability and monitors
LLM OP3
After exetting up
LANGCHAIN APT KEY
Inside the App we cons
Enry LLM Call, cost,
Latency, ip, off of LLM
is rendered in the
Long Smith Service URL
A N

Arrwer: Giren a task, agent executes a loop repeatedly to read stake until the goal Pick and run to somet Read results deude readsh Denplain were compounts of Lage (hain REACT agent LLM, tools, Prompt Structure) > LLM: Reamong engine behind any REACT agent > takes instructions in text > responds intext > Roduces thought > Arter Observation trace in each step. -> tools: - hiot of functions or callables. - Agent Picks on e of agriculture - If some payed by long and Olf observation 18 see promped by long Around text Around text Around text Around text Frompt Structure: Set of instructions Left scratch pod Few Shot example to thought Tool Acts Prompt + Scratch pod guide the converse to Keep reasons cha	(V)	what is an Agent?	
loop repeatedly to read state until the goal Pick and run to Read results dende nextsta dende nextsta dende nextsta dende nextsta REACT agat LLM, too's, Prompt Structure) > LLM: Reamong engine behind any REACT agat > takes instructions in text > responds in text > responds in text > responds in text > responds in text > recein each step. - too is: - List of functions or callables. - Agest Picks on e of anomabolic tools to po a given extra - Ips are poused by long and DIP observation is returned > Prompt Structure: Set of instructions Leman text Assulable tools Scratch pod Few Shot example for thought Tool Acts Prompt + Scratch pod Reap reasons cha			
Until the good Pick and run to (Somed) Read results dende neartist dende neartist dende neartist dende neartist Denplain core components of Long (hain REACT agent Structure) > LLM: Reanong engine behind any REACT agent > takes instructions in text > responds in text > Produces thought > Action Observation trace in each step. - Agent Picks on e of anomabolis tools to per a given extrain - Ips are poused by long and DIP observation is returned > Prompt Structure: Set of instructions Lems hot example for thought Thought For Shot example for thought For Shot example for thought Frompt + Scratch good Runde the converse to Keep reasons chan	Ar	• •	a
Read remits dende nexts+ dende nexts+ dende nexts+ dende nexts+ dende nexts+ Dende LLM, tools, Prompt Structure) > LLM: Rearong engine behind any REACT agent > takes instructions in text > responds intext > responds intext > responds intext > responds intext - responds intext - tools: _ List of functions or Callables. - Agent Picks one of anomable tools to per a given extra - Ips are powed by long and DIP observation is returned > Prompt Structure: Set of instructions Lems not exage from Adon Laboraryole Prompt + Scratch ped Rude the converse to keep rearong cha		loop repeatedly (to read state	
DENPlain core components of Lage (hair REACT agast (LLM, tools, Prompt Structure) > LLM: Reamong engine behind any REACT agast > takes instructions in text > responds in text > Produces thought > Action Observation trace in each step. - tools: - List of functions or Callables. - Agest Picks on e of anomable tools to per a given extran - Ips are powed by long and DIP observation is returned > Prompt Structure: Set of instructions Lember tools Scratch pod Four Shot example 1 thought Thought Frompt + Scratch pod Rocket tools Rude the converse to Keep reasons the		until the good) PICK and run to	loc
dende neadst Denplain core components of Lage (hair REACT agast Structure) > LLM: Reamong engine behind any REACT agast > takes instructions in text > responds in text > Produces thought > Action Observation trace in each step. - tools: - List of functions or callables. - Agest Picks on e of annihold tools to per a given extran - Ips are powed by long and DIP observation to re brings Annihold tools > Prompt Structure: Set of instructions Leman Hark Annihold tools Scratch pad Four Shot example -> Allon Loobserval P LLM dendes fool Acts Prompt + Scratchped guide the converse to Keep reasons cha			
REACT apol (LLM, too)s, Prompt Structure) -> LLM: Reamong engine behind any REACT agent -> take instructions in text -> responds intext -> Produces thought > Action Observation trace in each step. -> tools: - hiot of functions or Callables. Agent Picks on e of anomiable tools to per a given entiren Ips are poused by long and DIP observation 13 returned -> Prompt Structure: Set of instructions Lambde tools Scratch pod Four Shot example 1> thought Thought Thought Frompt + Scratch good guide the converse to Keep reasons cha		, .	tep.
REACT apol (LLM, too)s, Prompt Structure) -> LLM: Reamong engine behind any REACT agent -> take instructions in text -> responds intext -> Produces thought > Action Observation trace in each step. -> tools: - hiot of functions or Callables. Agent Picks on e of anomiable tools to per a given entiren Ips are poused by long and DIP observation 13 returned -> Prompt Structure: Set of instructions Lambde tools Scratch pod Four Shot example 1> thought Thought Thought Frompt + Scratch good guide the converse to Keep reasons cha	2)	Emploin core components of Lang (hois	^
Struture) -> LLM: Reavong engine behind any REACT agent -> takes instructions in text -> takes instructions in text -> responds intext -> Produces thought > Action objectivation trace in each step. -> tools: - List of functions or callables. - Agent Picks on e of animals tools to per a given entran - Ips are powed by long and DIP observation 13 returned -> Prompt Struture: Set of instructions Lambda tools Scratch pad Four Shot example -> thought -> Addon hobberrook P LLM dendes tool Acts Prompt + Scratchpod guide the converse to keep reasons cha			
-> LLM: Reasong engine behind any REACT agent > take instructions in text > responds intext > Produces thought > Action Observation trace in each step. -> tools: - hiot of functions or Callables. - Agent Picks on e of available tools to per a given eathor - Ips are poused by long and DIP observation is returned > Prompt Struture: Set of instructions Leman text Available tools Scratch pod Four Shot example for thought Thought Thought Plant deudes tool Acts Prompt + Scratch pod guide the converse to Keep reasons cha		· · · · · · · · · · · · · · · · · · ·	
any REACT agust Takes instructions in text Tespoords intext Produces thought > Autor Observator Trace in each step. Thoce in each step. Thoce in each step. Agent Picks one of availables. Agent Picks one of available tools to per a given eather The are persed by long and DIP observation The are persed by long and DIP observations The termod Prompt Structure: Set of instructions Lymain text Around tools Scratch pod Few Shot example Thought of thought Thought Tool Acts Prompt of Scratch pod Reep reasons cha		,	Δ
> take instructions in text > responds intext > Produces thought > Arter Observator trace in each step. - tools: - List of functions or Callables. - Agent Picks on e of a given extra - Ips are poused by long and DIP observation is returned > Prompt Struture: Set of instructions Lymain text Araulable tools Scratch pod Few Shot example for thought Lymain text Ly			>
> Produces thought > Action Observation trace in each step. -> tools: - List of functions or Callables. - Agent Picks one of anomable tools to per a given eather - Ips are powed by long and DIP observation 1s returned > Prompt Struture: Set of instructions Ly Main tank Anomable tools Scratch pod Four Shot example for thought Thought Plander Ly Main tank Anomable tools Scratch pod Four Shot example for thought Andon Lyobservation Reprompt + Scratch pod guide the correction to Keep reasons the			
> Produces thought > Action Observator trace in each step. -> tools: - List of functions or Callables. - Agent Picks on e of available tools to per a given eather - Ips are poured by long and OIP observation is returned > Prompt Struture: Set of instructions Lymain tears Analable tools Scratch pad Few Shot example for thought Tools Acts Prompt + Scratch pad guide the correction to keep reasons the			<u>, </u>
Dokervalor + trace in each step. -> tools: - List of functions or Callables. - Agest Picks on e of available tools to per a given extran - Ips are powed by long and DIP observation 13 xe torned -> Prompt Struture: Set of instructions Le Main tank Available tools Scratch pod Free Shot example for thought for thought Prompt of Scratch pod Rep reasong cha			
Callables. - Agest Picks on e of anomable tools to per a given entran - Ips are powed by long and DIP observation is returned > Prompt Struture: Set of instructions Lymain Lank Anomable tools Scratch pod Few Shot easys Lyndon Lybbarrole P LIM dendes tool Acts Prompt + Scratch pod Sude the converse to Keep yearong cha		> Produces thought > Action Observation	→
-> tools: - Liot of functions or Callables. - Agent Picks one of anomable tools to per a given extran - Ips are poused by long and DIP observation 13 returned >> Prompt Struture: Set of instructions Ly Main tearl Anomable tools Scratch pod Four Shot example 1> thoug it Ly Adon Lyobarrole P LLM dendes fool Acts Prompt + Scratch pod Sude the converse to Keep reasons the		trace in each step-	
Callables. - Agert Picks on e of anomable tools to per a given entran - Ips are poured by long and DIP observation 13 returned > Prompt Struture: Set of instructions Lymain tank Anomable tools Scratch pod Few Shot example 17 Adon Lyobarroal PLIM dendes tool Acts Prompt + Scratch pod Sude the converse to Keep reasons cha			
a given entran - If so are poused by long ord DIP abservation is returned > Prompt Struture: Set of instructions Lymain tank Anoutable tooly Scratch pad Four Shot example for thought Lymber rate Prompt + Scratch pad Sude the converse to Keep reasons the		· ·	
a given entran - If so are poused by long ord DIP abservation is returned > Prompt Struture: Set of instructions Lymain tank Anoutable tooly Scratch pad Four Shot example for thought Lymber rate Prompt + Scratch pad Sude the converse to Keep reasons the		- Agest Picks one of	
a given entran - Ips are poused by long and DIP observation is returned > Prompt Struture: Set of instructions Le Main Hark Anoughle tooly Scratch pad Few Shot example Le thouse M Le thouse M Le thouse M Prompt + Scratch pad guide the converse to Keep reasons cha		v	
- Ips are poused by long and OIP observation is returned > Prompt Struture: Set of instructions Ly Main tank Assulable tooly Scratch pad Four Shot example for thought Tobrervation Lyobrervation Prompt + Scratch pad guide the converse to Keep reasons cha			
ord DIP observation 13 returned > Prompt Struture: Set of instructions Lymain Lank Aroutable tools Scratch pad Few Shot except for thought Tobrerrate Prompt + Scratch pad Reep reasons char Reep reasons char 13 returned			ષ્ વે
Prompt Struture: Set of instructions Limbin tears Available tools Scratch pad Four Shot example for thought Tobar rate Prompt + Scratch pad guide the converse to Keep reasons chain		,	
> Prompt Struture: Set of instructions Lymain tears Aroutable tools Scratch pad Four Shot evangs for thought Arobert value PLLM dendes Tool Acts Prompt + Scratch pad Swde the converse to Keep reasons cha			
Set of instructions Le Main tenk Available tools Scratch pod Few Shot evange Les thought Thomas M T			
Asomble tools Scratch pad Four Shot every fre though Though Looker role Lool Acts Prompt + Scratch pad guide the converse to Keep reasons cha		•	
Available tools Scratch pad Four Shot evage from though Th		1500 100 P	
Scratch pad Few Shot example from Shot example from Adorn hobby role PLIM dendes tool Acts Prompt + Scratch pad guide the converse to Keep reasons chan		,	
Few Shot early from Shot early from Adon hobserval tool Acts Prompt + Scratchpod gwise the converse to keep reasons cha		/	
Frompt + Scratchped ? Sweethe Converse to Keep reasons cha		·	
PLLM dendes tool Acts Prompt + Scratchpod) guide the converse to Keep reasons cha		Few Shot except	نعع
PLLM dendes tool Acts Prompt + Scratchpod guide the converse to Keep reasons cha		to thouse	₹
PLLM dendes tool Acts Prompt + Scratchpod gwde the converse to Keep reasons cha		,	
Prompt + Scratchpody guide the converse to Keep reasons cha			
Prompt + Scratchpody 8 u-de the converso to Keep reasons cha		A	
gude the converse to Keep reasons cha		^	
gude the converse to Keep reasons cha		rompt + Scratchpod	
•		guide the Convers	stą
•		40	
		•	
transparent, 1 tera		transparent, i ter	ate

> Whateve w	re buildug '!		
REACT 1	AgetExecut	for from o	Scrotch
	0	·	