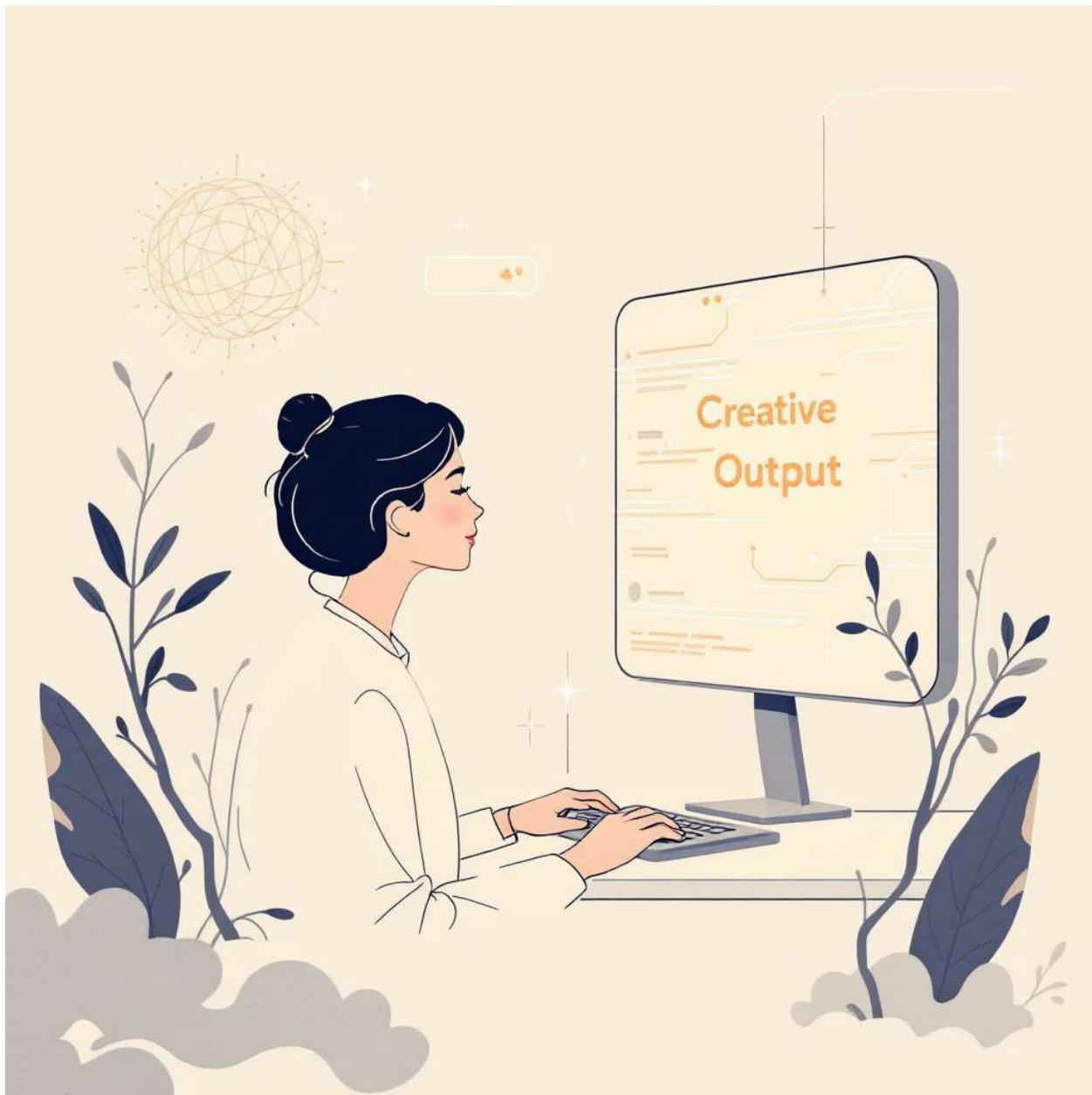


جزوه پرامپت‌نویسی (مهندسی پرامپت به زبان خودمانی)



فهرست مطالب

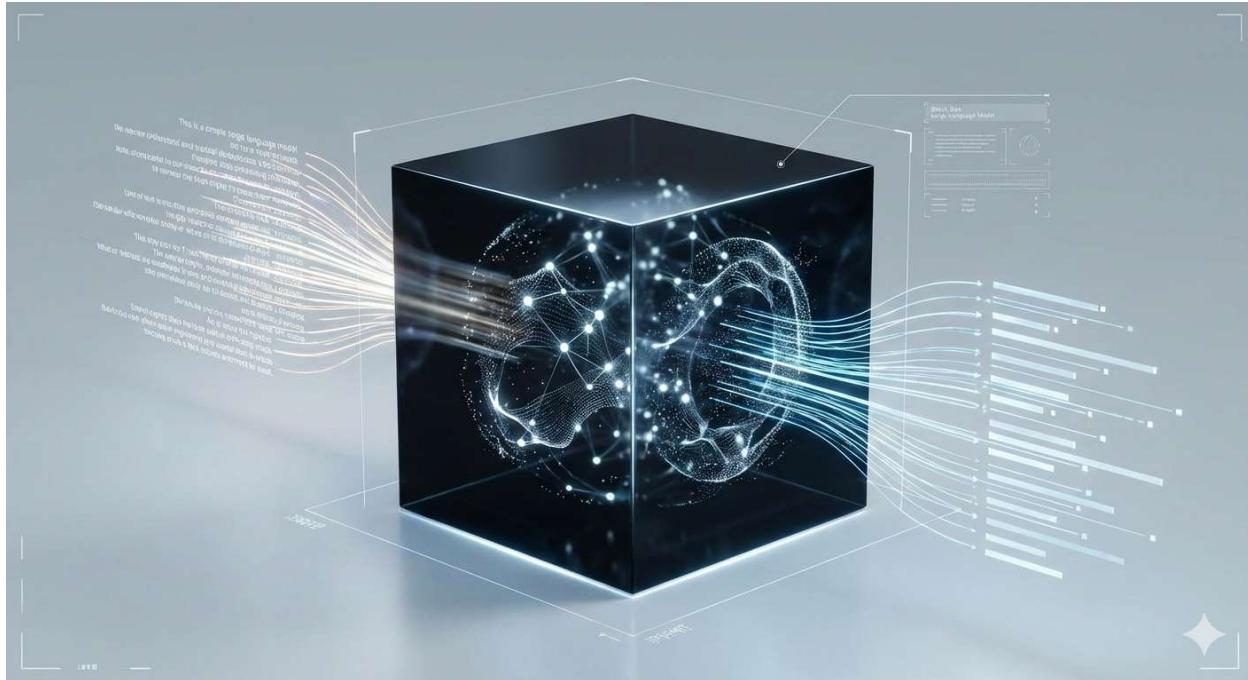
۳.....	مقدمه	•
	○ پرامپت‌نویسی: چت کردن یا برنامه‌نویسی با زبان طبیعی؟	
۴.....	جعبه سیاه LLM و نگاه اولیه	•
	○ پشت پرده مدل‌های زبانی بزرگ (مغز آماری و پیش‌بینی کلمه بعدی)	
	○ چرا خروجی‌ها متفاوت‌اند؟ (Sampling و احتمالات)	
۵.....	پرامپت‌نویسی به عنوان یک هنر	•
۶.....	کنترل خروجی‌ها	•
	○ مسئله: چطور خروجی کمتر تصادفی باشد؟	
	○ مفهوم Temperature (دماهی تولید)	
	○ کنترل از طریق متن پرامپت (دقیق vs آزاد)	
	○ نقش حافظه کوتاه‌مدت در گفتگو	
۸.....	تعادل بین خلاقیت و کنترل	•
	○ استعاره رانندگی: آزادی کامل vs محدودیت کامل	
	○ راهکارهای ایجاد تعادل (چارچوب، سبک، هدف)	
۹.....	ساخت مرحله‌به‌مرحله پرامپت	•
	○ روند مرحله‌ای (نگاه ساده)	
	○ گذر نرم به نگاه حرفاًی (پل بین ساده و استاندارد)	
۱۱.....	اسکلت حرفاًی پرامپت (نگاه استاندارد)	•
	○ جدول اجزای اصلی (Role, Instruction, Context, ...)	
	○ نمونه‌ی ترکیبی (مقایسه نگاه ساده و حرفاًی)	
۱۴.....	اصول طلایی (فوت‌های کوزه‌گری)	•
	○ شفافیت و جزئیات	
	○ قدرت مثال‌ها (Few-shot)	
	○ تفکر گام‌به‌گام و زنجیره‌ای	
۱۶.....	سخن پایانی	•

مقدمه

پرامپت‌نویسی: چت کردن یا برنامه‌نویسی با زبان طبیعی؟

از یک مقدمه شروع کنیم بینیم چی به چیه و به کجا میخوایم برسیم.

وقتی می‌گیم پرامپت می‌نویسیم و خروجی می‌گیریم، در واقع داریم با یک جعبه سیاه کار می‌کنیم. این جعبه سیاه همون **LLM** هست. تو فقط ورودی میدی، خروجی می‌گیری، ولی اینکه داخلش چی می‌گذرد رو نمی‌بینی.



جعبه سیاه LLM و نگاه اولیه

پشت پرده مدل‌های زبانی بزرگ (مغز آماری و پیش‌بینی کلمه بعدی)

حالا پشت این ماجرا چی هست؟

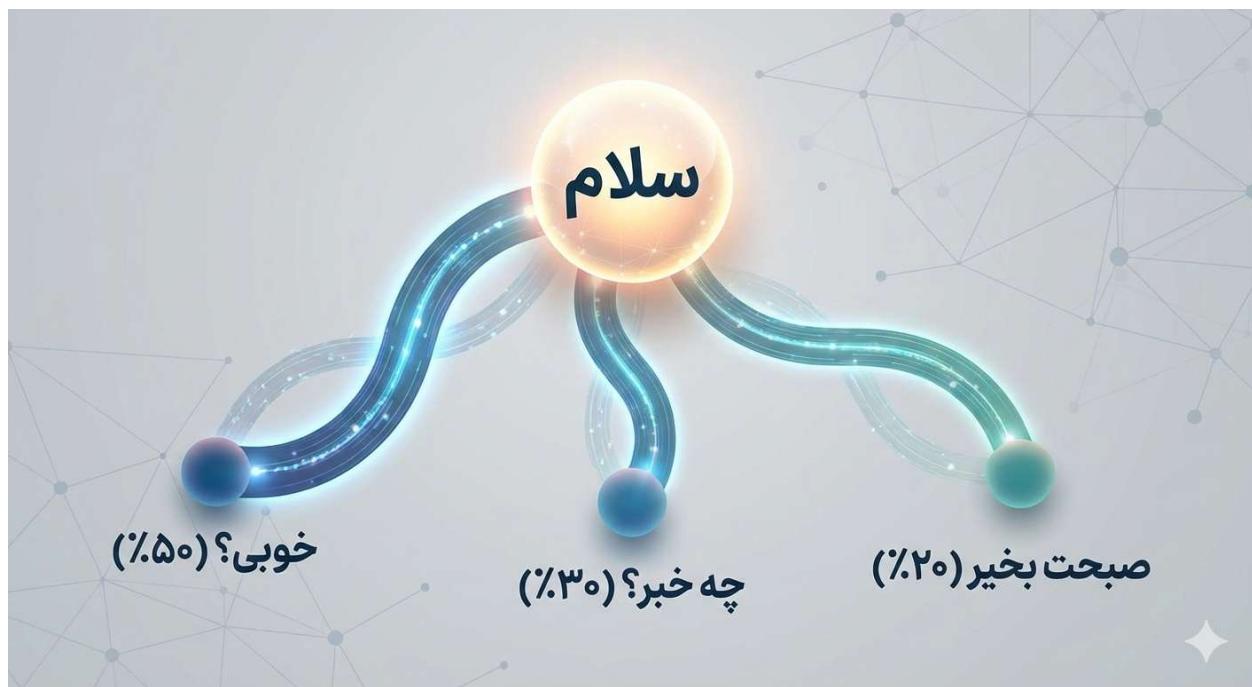
مدل‌های زبانی بزرگ مثل یه مغز آماری عمل می‌کنن. یعنی چی؟ یعنی میلیارداها کلمه و جمله رو قبل‌اً دیده و یاد گرفته که وقتی یه رشته‌ای از کلمات میاد، احتمال اینکه کلمه‌ی بعدی چی باشد را حساب کنه. پس وقتی تو پرامپت میدی، مدل شروع می‌کنه کلمه به کلمه پیش‌بینی کردن.

چرا خروجی‌ها متفاوت‌اند؟ (Sampling و احتمالات)

اینکه چرا برای یه ورودی مشابه خروجی متفاوت می‌گیریم، دلیلش اینه که مدل همیشه یه جور قطعیت نداره. اون احتمال‌ها رو مثل یه نقشه می‌بینه:

مثلاً می‌دونه بعد از "سلام" می‌تونه "خوبی؟" بیاد یا "چه خبر؟".

هر بار ممکنه یکی از این مسیرها رو انتخاب کنه. این انتخاب هم به خاطر چیزی به اسم **Sampling** یا نمونه‌گیریه. یعنی مدل از بین گزینه‌های محتمل یکی رو برمی‌داره، و همین باعث میشه خروجی‌ها متنوع باشن.



پرامپت‌نویسی به عنوان یک هنر



پس پشت این داستان، چیزی خیلی پیچیده‌تر از یه تابع ساده هست. تو داری به یه سیستم آماری عظیم ورودی میدی، اون هم بر اساس احتمالات و تجربه‌ی قبلیش خروجی میده. به همین خاطر پرامپت‌نویسی شبیه یه هنر میشه: تو باید یاد بگیری چطور ورودی رو طوری بچینی که مدل رو به سمت خروجی‌ای که می‌خوای هدایت کنه.

کنترل خروجی‌ها

مسئله: چطور خروجی کمتر تصادفی باشد؟

خب حالا مسئله رو اینطوری بیان می‌کنیم: چطور میشه پرامپت رو کنترل کرد تا خروجی کمتر تصادفی باشه؟ وقتی گفتیم خروجی‌ها می‌تونن متفاوت باشن، حالا چطور میشه این تفاوت رو کنترل کرد؟

مفهوم **Temperature** (دماهی تولید)

مدل‌ها یه چیزی دارن به اسم **Temperature** یا همون دماهی تولید. اینو مثل یه پیچ تنظیم تصور کن:

- وقتی دما پایینه: مدل محافظه‌کار میشه و بیشتر میره سمت جواب‌های قطعی و تکراری. یعنی خروجی‌ها شبیه هم میشن.
- وقتی دما بالاست: مدل خلاق‌تر میشه و احتمال‌های عجیب‌تر رو هم امتحان می‌کنه. اینجا خروجی‌ها متنوع‌تر میشن.



کنترل از طریق متن پرامپت (دقیق vs آزاد)

پس یکی از راههای کنترل همین تنظیمات پشت پرده است. اما از دید پرامپتنویسی، تو می‌تونی با خود متن ورودی هم این کنترل رو اعمال کنی.

مثلاً وقتی خیلی دقیق و محدود پرامپت می‌دی، مدل کمتر راه فرار داره و خروجی‌ها نزدیک‌تر به هم میشن. ولی وقتی پرامپت آزاد و کلی میدی، مدل می‌تونه هزار جور مسیر بره و خروجی‌ها متفاوت‌تر میشن.

نقش حافظه کوتاه‌مدت در گفتگو

یه نکته‌ی دیگه هم اینه که مدل‌ها حافظه‌ی کوتاه‌مدت دارن. یعنی توی همون گفتگو، چیزی که قبلاً گفتی روی خروجی بعدی اثر می‌ذاره. پس پرامپتنویسی فقط یه ورودی جدا نیست، بلکه یه جریان گفتگویه که تو می‌تونی با مدیریت این جریان، خروجی رو شکل بدی.

مثال ساده: داستان گربه آزاد vs داستان گربه سیاه در کتابخانه

می‌خوای یه مثال خیلی ساده بزنم که بینیم چطور با تغییر لحن یا جزئیات پرامپت، خروجی از حالت تصادفی به سمت قابل پیش‌بینی‌تر میره؟

پس بریم جلو و روند رو کامل کنیم. گفتیم که خروجی‌ها می‌تونن متفاوت باشن و ما می‌تونیم با دما یا جزئیات پرامپت کنترل‌ش کنیم. حالا بذار یه مثال خیلی ساده بزنم:

۱. پرامپت آزاد:

«یک داستان کوتاه درباره‌ی یک گربه بنویس.»

اینجا مدل می‌تونه هزار جور مسیر بره: گربه‌ای که توی خیابون زندگی می‌کنه، گربه‌ای که با بچه‌ها بازی می‌کنه. چون پرامپت آزاد و کلیه.

۲. پرامپت دقیق:

«یک داستان کوتاه درباره‌ی یک گربه‌ی سیاه که در یک کتابخانه‌ی قدیمی زندگی می‌کند و شب‌ها کتاب‌ها را می‌خواند.»

اینجا مدل خیلی کمتر راه فرار داره. خروجی‌ها شبیه‌تر میشن، چون تو مسیر رو مشخص کردی.

جمع‌بندی این بخش:

این یعنی پرامپتنویسی مثل هدایت کردن مدل به سمت خروجی دلخواه‌ته. هرچی دقیق‌تر و محدود‌تر باشی، خروجی قابل پیش‌بینی‌تر میش. هرچی آزادتر باشی، خروجی خلاق‌تر ولی متنوع‌تر میش.

پس روند کلی اینه:

۱. بفهمیم مدل چطور کار می‌کنه (پیش‌بینی کلمه بعدی بر اساس احتمال).

۲. بفهمیم چرا خروجی‌ها متفاوت میشن (sampling و دما).

3. یاد بگیریم چطور با پرامپت دقیق یا آزاد، خروجی را کنترل کنیم.

تعادل بین خلاقیت و کنترل

چطور میشه پرامپت رو طوری طراحی کرد که هم خلاقیت داشته باشه و هم کنترل؟

استعاره رانندگی: آزادی کامل vs محدودیت کامل

این همون تعادل جذاب بین آزادی و محدودیت در پرامپت‌نویسیه. مثل رانندگی: اگر همه‌ی خیابون‌ها باز باش، آزادی داری ولی احتمال گم شدن زیاده؛ اگر خیلی محدود باش، فقط یک مسیر داری و خلاقیت می‌میره. پس باید یک جاده‌ی نیمه‌باز سازی: جهت‌دار اما با فضای مانور.



راهکارهای ایجاد تعادل

• چارچوب بده و جا برای بازی بذار:

○ مثال: «یک داستان کوتاه درباره‌ی یک گربه‌ی سیاه در کتابخانه‌ی قدیمی، با پایان غافلگیر کننده.»

○ محدودیت‌ها مشخص‌اند (موضوع، مکان، ویژگی) و آزادی هم هست (نوع غافلگیری).

• سبک یا لحن را تعیین کن:

◦ مثال: «به زبان طنز» یا «به شکل شاعرانه».

◦ کنترل بیشتر می‌گیری، اما همچنان مسیرهای خلاق باز می‌مانند.

• هدف را روشن کن:

◦ آموزش؟ سرگرمی؟ توضیح علمی؟ وقتی هدف روشن باشد، خروجی به خواسته‌ات نزدیک‌تر می‌شود و از پراکندگی جلوگیری می‌کند.

• قرارداد بین تو و مدل:

◦ این قرارداد کوچک بین تو و مدل مشخص می‌کند چه چیزهایی قطعی باشند و کجاها آزاد، تا خروجی هم قابل پیش‌بینی باشد و هم تازه و جذاب.

ساخت مرحله به مرحله پرامپت

برای حفظ ریتم روان، اول با نگاه ساده جلو می‌رویم، بعد بی‌دردسر وارد نسخه‌ی حرفا‌ی می‌شویم. این «پل» کمک می‌کند خواننده حس نکند ناگهان به زبان تخصصی پرت شده.

نگاه ساده (روند مرحله‌ای)

1. موضوع کلی را بگو: «یک داستان کوتاه درباره یک گربه.»

2. جزئیات زمینه را اضافه کن: «گربه‌ی سیاه که در کتابخانه‌ی قدیمی زندگی می‌کند.»

3. سبک یا لحن را مشخص کن: «به زبان طنز.»

4. هدف یا نتیجه را تعیین کن: «پایان غافلگیرکننده باشد.»

5. محدودیت یا قالب بده: «کمتر از 200 کلمه.»

این پنج قدم مثل دستور آشپزی‌اند: هر لایه مسیر را روشن‌تر و خروجی را قابل کنترل‌تر می‌کند.



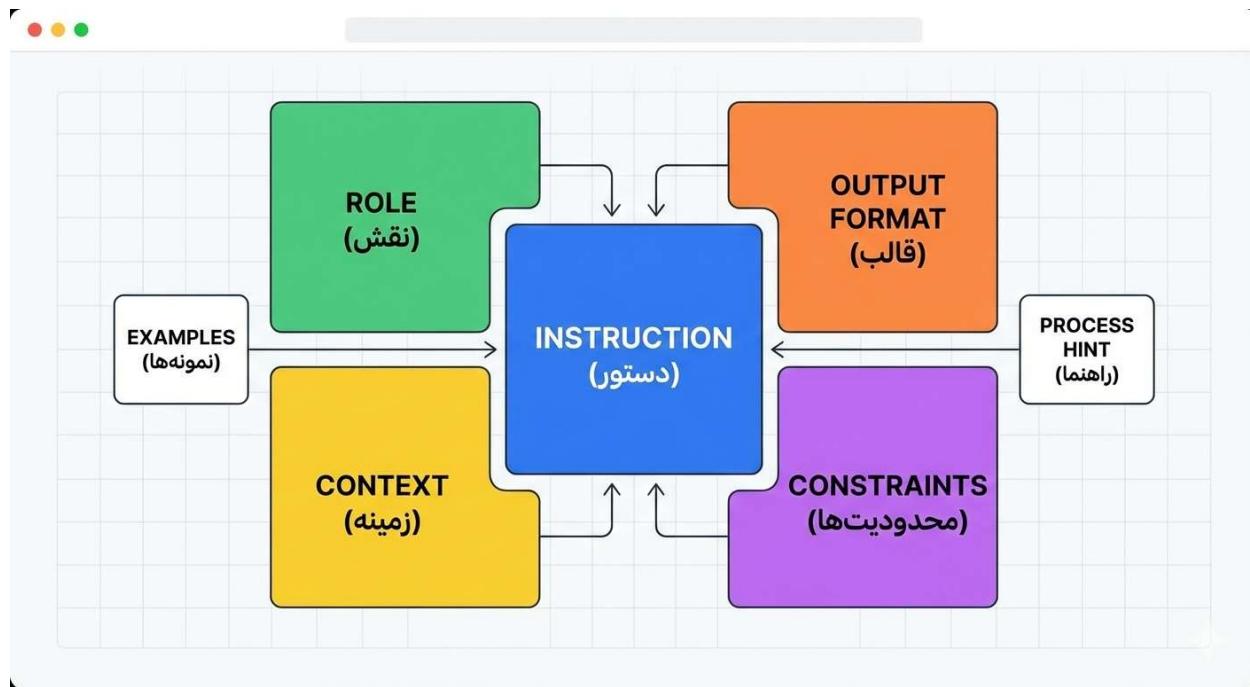
گذر نرم به نگاه حرفه‌ای (پل بین ساده و استاندارد)

برای اینکه گذار طبیعی بماند، همین پنج قدم را به معادل‌های حرفه‌ای‌شان وصل می‌کنیم:

- دستور(**Instruction**): همان «موضوع کلی و کاری که باید انجام شود» را دقیق و یکخطی می‌کند.
 - زمینه(**Context**): همان «جزئیات زمینه» است که مخاطب، شرایط و پس‌زمینه را روشن می‌سازد.
 - محدودیت یا نقش (**Role Constraints** یا **Role Constraints**): همان «سبک/لحن» است، یا هویت پاسخ‌دهنده را تعیین می‌کند تا خروجی منسجم شود.
 - محدودیت و معیار (**Constraints & Criteria**): همان «هدف/نتیجه» است که شاخص‌های مطلوب را مشخص می‌کند.
 - قالب خروجی (**Output Format**): همان «حدودیت یا قالب» است که شکل تحویل را استاندارد می‌کند.
- با این نگاشت، خواننده همان چیزی را که فهمیده، فقط به زبان استاندارد می‌بیند؛ پس انسجام روایت حفظ می‌شود.

اسکلت حرفه‌ای پرامپت (نگاه استاندارد)

قبل از فهرست، یک نکته‌ی راهگشا: اسکلت حرفه‌ای کمک می‌کند همان قصد سادهات را به زبان دقیق و قابل‌اجرا برای مدل تبدیل کنی. هر بند یک پیج تنظیم است؛ با سفت‌کردن یا شل‌کردن این پیج‌ها، تعادل بین کنترل و خلاقیت را دقیق‌تر می‌سازی.



بخش (Component)	مثال	توضیح کوتاه
Role (نقش)	Act as a creative storyteller / به عنوان نویسنده طنز	نقش به مدل زاویه‌ی دید و مسئولیت می‌دهد؛ مثل انتخاب لنز دوربین.
Instruction (دستور)	Write a short story about a black cat living in an old library	دستور باید شفاف و قابل سنجش باشد تا مدل مسیر را گم نکند.
Context (زمینه)	Audience: beginner readers; tone: humorous; setting: mysterious library	زمینه مرزها و محیط را مشخص می‌کند؛ از پراکنده‌گویی جلوگیری می‌کند.
Output Format (قالب)	.words, single paragraph $200 \geq$	قالب خروجی، شکل تحويل را استاندارد می‌کند تا مقایسه و استفاده آسان شود.
Examples (نمونه‌ها)	...!Input: ... / Output	نمونه‌ها الگو می‌دهند؛ مدل سبک و ریتم را از آن‌ها یاد می‌گیرد.
Constraints (حدودیت‌ها)	.No repetition, keep it light, surprising ending	حدودیت‌ها کیفیت را نگه می‌دارند و از حاشیه‌روی جلوگیری می‌کنند.
Process Hint (راهنمای)	Reason step-by-step before writing the final .story	راهنمای فرایند به مدل می‌گوید چطور فکر کند؛ مخصوصاً در کارهای پیچیده.

نمونه‌ی ترکیبی (садه + حرفه‌ای)

یک نمونه جمع‌وجور که هر دو نگاه را پیوند می‌دهد:

نگاه ساده:

«یک داستان کوتاه طنز درباره‌ی یک گربه‌ی سیاه در کتابخانه‌ی قدیمی، با پایان غافلگیرکننده، کمتر از 200 کلمه.»

نگاه حرفه‌ای:

Role: Act as a creative storyteller.

Instruction: Write a humorous short story about a black cat living in an old library.

Context: Audience is general readers; keep the setting slightly mysterious.

Output format: ≤ 200 words, single paragraph.

Constraints: No fluff; surprising ending; light tone.

Process hint: First outline the plot beats, then write the final paragraph.

چند اصل طلایی (فوت‌های کوزه‌گری)

خب حالا بريم چندتا اصل رو ياد بگيريم که اگر همین امروز رعایت کني، خروجي هات جهش مي کنه. هر کدوم رو با توضیح کامل و مثال باز می کنیم:

۱. **شفاف و مشخص بنویس** وقتی پرامپت مبهم باشه، مدل هم خروجی مبهم می ده. مثل اینه که به کسی بگی «یه چیزی بگو»؛ خوب هزار جور جواب ممکنه بده. ولی اگر دقیق بگی «یک خلاصه‌ی ۳ پاراگرافی از این متن برای دانشجوی مبتدی، به زبان ساده»، خروجی خیلی نزدیکتر به خواسته میشه.

• مثال مبهم: «یک خلاصه بده.» (مدل ممکنه خلاصه‌ی خیلی کوتاه یا خیلی طولانی بده، یا حتی سبک متفاوتی انتخاب کنه).

• مثال شفاف: «یک خلاصه‌ی ۳ پاراگرافی از این متن بده، به زبان ساده، برای دانشجوی مبتدی، در کمتر از ۲۰۰ کلمه.»

۲. از مثال استفاده کن (تکنیک **Few-shot**) مدل‌ها با مثال خیلی بهتر یاد می‌گیرن. وقتی یک نمونه ورودی-خروجی بهش می‌دی، سبک و قالب رو کپی می‌کنه. این تکنیک رو بهش می‌گن **Few-shot prompting**.

• مثال بدون نمونه: «یک متن تبلیغاتی برای یک محصول جدید بنویس.» (خروجی ممکنه رسمی، غیررسمی یا خیلی کلی باشه).

• مثال با نمونه: «یک متن تبلیغاتی برای یک محصول جدید بنویس. نمونه‌ی خوب: محصول: گوشی هوشمند / متن: "با سرعتی فراتر از انتظار، گوشی جدید X تجربه‌ای متفاوت برای شما می‌سازد." حالا برای یک لپ‌تاپ جدید متن مشابه بده.» (مدل سبک تبلیغاتی رو یاد می‌گیره و خروجی همون قالب رو دنبال می‌کنه).

۳. گام‌به‌گام درخواست کن (تکنیک **Chain of Thought**) وقتی کار پیچیده‌ست، اگر همه رو یکجا بگی، مدل ممکنه گیج بشه یا جواب نصفه بده. ولی اگر مرحله‌به‌مرحله بخوای، خروجی دقیق‌تر میشه. این تکنیک رو **Chain of Thought** (یا همون فکر کردن زنجیره‌ای خودمون) می‌گن.

• مثال یکجا: «یک مقاله درباره‌ی تاریخچه‌ی اینترنت بنویس.»

• مثال گام‌به‌گام: «اول یک فهرست از مهم‌ترین رویدادهای تاریخ اینترنت بده. بعد برای هر رویداد یک توضیح ۳ جمله‌ای بنویس. در آخر یک جمع‌بندی کوتاه بده.» (خروجی مرحله‌به‌مرحله ساخته میشه و کامل‌تره).

۴. زنجیره‌ای کار کردن (تکنیک **Prompt Chaining**) گاهی بهتره خروجی یک مرحله رو ورودی مرحله‌ی بعد کنی. این کار باعث میشه مدل روی نتایج خودش فکر کنه و کیفیت بالا بره.

نقشه راه زنجیره‌ای (مثال تولید مقاله):

• گام ۱ (ایده‌پردازی):

ورودی: «۱۰ تا ایده برای مقاله در مورد هوش مصنوعی بدء.»

خروجی: (لیستی از ۱۰ ایده خام)

(این خروجی میره مرحله بعد)

• گام ۲ (گلچین کردن):

ورودی: «از بین این لیست بالا، ۳ تا که جذاب‌ترن رو جدا کن.»

خروجی: (۳ عنوان برتر)

(این خروجی میره مرحله بعد)

• گام ۳ (نوشتن نهایی):

ورودی: «حالا برای هر کدوم از این ۳ تا، یک مقدمه بنویس.»

خروجی: (محتوای آماده و باکیفیت)

چرا این کار رو کردیم؟ اگر یک دفعه می‌گفتی «یه مقاله بنویس»، مدل گیج می‌شد و کیفیت پایین می‌ومد. اما با این زنجیره، توی هر مرحله تمرکز مدل رو روی یه کار خاص گذاشتی.

۵. آزمون و خطا هیچ پرامپتی از اول عالی نیست. باید چند بار امتحان کنی، تغییر بدی، و نسخه‌های مختلف رو ذخیره کنی. این کار مثل طراحی آزمایشی یک محصوله: هر بار یک متغیر رو تغییر بدی تا بفهمی چه چیزی بهتر جواب می‌ده.

• مثال تکامل پرامپت:

○ نسخه ۱: «یک داستان کوتاه درباره‌ی یک گربه.»

○ نسخه ۲: «یک داستان کوتاه طنز درباره‌ی یک گربه‌ی سیاه.»

○ نسخه ۳: «یک داستان کوتاه طنز درباره‌ی یک گربه‌ی سیاه در کتابخانه‌ی قدیمی، با پایان غافلگیر‌کننده.»

جمع‌بندی

این اصول مثل یک جعبه‌ابزار برای پرامپت‌نویسی هستن. اگر شفاف بنویسی، مثال بدی، مرحله‌به‌مرحله کار کنی، زنجیره‌ای پیش ببری و آزمون و خطا انجام بدی، خروجی‌هات هم قابل کنترل‌تر میشن و هم خلاق‌تر.

سخن پایانی

از کجا به کجا رسیدیم؟

خوب دمتونگرم که ٿا آخِر این چیزه پا من همراه پودی، ما سقدمون رو از شناخت اون "جعیبه سیاه" آماری شروع کردیم و فهمیدیم که LLM ها چادو گر نیستن؛ اوئی فقط پیش پیش پیش پیش کنندہ های خیلی پاھوڻ کلمه ها هستن که پا احتمالات کار می کنن. یاد گرفتیم که چطور پا ترتیبیم «دما» و دقیق تر کردن در حواسِ هامون، افسار این مدل ها رو دستمون پکیدیم و از حالت «چیز کردن معمولی» په سمت «پرنامه نویسی پا زبان طبیعی» حرکت کنیم.

توی مسیر، یاد گرفتیم که یه پرامپت حرفه ای مثل یه ساختمن، اسکلت (Role, Context, Instruction) داره و پا فوٹ های کوژه گری مثل مثال زدن (Few-shot) و فکر کردن گام په گام، می شه خروجی های اعجاب انگیزی ارزش گرفت. در نهایت هم پا هم تمرین کردیم و دیدیم که ھیچ پرامپت کاملی و چوڈ نداره، پلکه همه چیز توی آزمون و خطوا ساختن اون کتابخانه ای شخصی پرامپت ها پخته می شه.

حالا دیگه تو پوچ تو زمین توئه! یادت پاشه پرامپت نویسی یه مهارت که پا هر بار تایپ کردن، توی وجودت عمیق تر می شه. از این په بعد، هر بار که حواسِ ها پا هوش مصنوعی کار کنی، په چای یه دستور ساده، په اون چارت مهندسی و نقش هایی که یاد گرفتی فکر کن، پدو چلو، اشتپاہ کن، اصلاح کن و خروجی هایی پکید که پقیه رو شکعت زده کنه. مسیر مهندس پرامپت شدن تو ٿا زره از اینجا شروع شده؛ موافق پاشی!