جزوه کامل فارسی

راهنمای پرامپت‌نویسی GPT-5.5

بر اساس مستندات رسمی OpenAI API - Prompt guidance

نسخه آموزشی، ساختارمند و کاربردی برای طراحی پرامپت‌های حرفه‌ای

تاریخ تهیه: ۱ ژوئن ۲۰۲۶

نکته: این جزوه برای استفاده آموزشی و اجرایی نوشته شده و بخش‌های اصلی راهنمای رسمی را به زبان فارسی توضیح می‌دهد؛ مثال‌ها و قالب‌های فارسی برای کاربرد مستقیم در پروژه‌ها اضافه شده‌اند.

فهرست مطالب

مقدمه: این جزوه درباره چیست؟

این جزوه نسخه فارسی، منظم و آموزشیِ راهنمای رسمی OpenAI برای پرامپت‌نویسی با GPT-5.5 است. هدف آن این است که نکات صفحه Prompt guidance را به زبان فارسی، با توضیح کاربردی، مثال و الگوی آماده، در قالبی قابل استفاده برای یادگیری، پیاده‌سازی در محصول و ساخت prompt stackهای حرفه‌ای ارائه کند.

پیام اصلی راهنما این است: در GPT-5.5 به جای اینکه مسیر فکرکردن مدل را با دستورهای طولانی و قدم‌به‌قدم کنترل کنیم، باید مقصد، معیار موفقیت، محدودیت‌ها، منابع شواهد، شکل خروجی و قوانین توقف را واضح تعریف کنیم. مدل در این حالت بهتر می‌تواند مسیر مؤثر را خودش انتخاب کند.

این متن ترجمه خط‌به‌خط رسمی نیست؛ بلکه یک جزوه فارسیِ وفادار به محتوا و نکات راهنماست. برای دسترسی به متن رسمی و به‌روز، لینک منبع اصلی در پایان جزوه آمده است.

۱. تازه‌های GPT-5.5 نسبت به GPT-5.4

OpenAI در ابتدای راهنما چهار تغییر اصلی را برجسته می‌کند. نخست اینکه پرامپت‌های کوتاه‌تر و نتیجه‌محور معمولاً از prompt stackهای سنگین و فرایندمحور بهتر عمل می‌کنند. دوم اینکه چون reasoning مدل کارآمدتر شده، قبل از بالا بردن reasoning effort باید دوباره بررسی کرد که آیا حالت‌های low یا medium کافی هستند یا نه.

سوم اینکه در workflowهای سنگین با ابزار، مفاهیمی مثل preamble، مدیریت phase و replay صحیح assistant itemها همچنان مهم هستند. چهارم اینکه برای تجربه‌های کاربرمحور و agentic، تعریف شخصیت، بودجه بازیابی اطلاعات و قوانین validation کمک می‌کند رفتار مدل قابل‌کنترل‌تر و قابل‌اعتمادتر شود.

نتیجه عملی: وقتی یک پرامپت قدیمی را به GPT-5.5 منتقل می‌کنید، لازم نیست تمام دستورهای قدیمی را همان‌طور نگه دارید. دستورهای بیش از حد ریز، مخصوصاً دستورهایی که مسیر را به جای مقصد کنترل می‌کنند، ممکن است باعث نویز، محدود شدن فضای جست‌وجوی مدل یا خروجی‌های مکانیکی شوند.

۲. اصل مرکزی: مقصد را تعریف کن، نه تمام مسیر را

GPT-5.5 وقتی بهتر کار می‌کند که پرامپت به مدل بگوید خروجی خوب چه شکلی است، چه محدودیت‌هایی مهم‌اند، چه شواهدی در دسترس است و پاسخ نهایی باید شامل چه چیزهایی باشد. در چنین حالتی مدل آزادی دارد که بهترین مسیر، ابزار یا استراتژی reasoning را انتخاب کند.

به جای نوشتن دستورهایی مثل «اول A را بررسی کن، بعد B را بررسی کن، بعد تمام استثناها را تحلیل کن، بعد ابزار X را بزن»، بهتر است بنویسیم: «مسئله کاربر را تا انتها حل کن؛ موفقیت یعنی تصمیم درست با داده‌های موجود گرفته شود، اقدام مجاز انجام شود، پاسخ نهایی شامل اقدامات انجام‌شده و موانع باشد، و اگر شواهد کافی نیست فقط کوچک‌ترین داده ضروری پرسیده شود.»

این تغییر ظاهراً ساده است، اما برای کیفیت پاسخ بسیار مهم است. پرامپت نتیجه‌محور به مدل می‌گوید چه چیزی مهم است؛ پرامپت فرایندمحور اغلب مدل را وادار می‌کند حتی وقتی مسیر کوتاه‌تر و دقیق‌تری وجود دارد، یک سلسله‌مراتب ثابت و گاهی غیرضروری را اجرا کند.

۳. استفاده دقیق از واژه‌های مطلق مثل ALWAYS و NEVER

در راهنما تأکید شده که واژه‌هایی مثل must، NEVER، ALWAYS و only باید فقط برای قواعد واقعاً ثابت و غیرقابل‌مذاکره استفاده شوند؛ مانند قوانین ایمنی، فیلدهای اجباری خروجی، یا اقداماتی که هرگز نباید انجام شوند.

برای تصمیم‌هایی که به موقعیت وابسته‌اند، مثل اینکه مدل چه زمانی جست‌وجو کند، چه زمانی ابزار بزند، چه زمانی سؤال بپرسد یا چه زمانی ادامه دهد، بهتر است قانون تصمیم‌گیری تعریف شود. مثال بد: «همیشه سرچ کن.» مثال بهتر: «وقتی اطلاعات ممکن است تغییر کرده باشد، وقتی منبع لازم است، یا وقتی بدون جست‌وجو احتمال خطا وجود دارد، سرچ کن.»

این تفاوت باعث می‌شود مدل به جای اطاعت کورکورانه از یک دستور مطلق، بین دقت، سرعت، هزینه و نیاز واقعی کاربر تعادل برقرار کند.

۴. قوانین توقف: چه زمانی ادامه بدهیم و چه زمانی پاسخ بدهیم؟

یکی از نکات مهم GPT-5.5 تعریف stopping conditions است. هدف این نیست که مدل همیشه کمترین تعداد ابزار یا کمترین مرحله را استفاده کند؛ هدف این است که با کمترین حلقه مفید به پاسخ درست برسد، بدون اینکه صحت، شواهد، محاسبات یا citationهای لازم قربانی شوند.

یک قانون توقف خوب این است: بعد از هر نتیجه یا ابزار، مدل بررسی کند آیا اکنون می‌تواند درخواست اصلی کاربر را با شواهد کافی و citationهای لازم پاسخ دهد یا نه. اگر بله، باید پاسخ بدهد. اگر نه، فقط همان مرحله‌ای را ادامه دهد که برای تکمیل پاسخ ضروری است.

در نبود چنین قانونی، مدل ممکن است یا زودتر از موعد متوقف شود و پاسخ ناقص بدهد، یا برعکس در حلقه‌های غیرضروری جست‌وجو و بررسی بماند.

۵. رفتار در نبود شواهد کافی

در پاسخ‌های grounded یا مبتنی بر منبع، نبود شواهد نباید خودکار به معنی «نه»، «وجود ندارد» یا «قطعاً غلط است» تفسیر شود. رفتار درست این است که مدل بگوید: «در منابع بررسی‌شده شواهد کافی پیدا نشد» یا «بر اساس داده‌های موجود نمی‌توان با قطعیت گفت.»

راهنمای OpenAI پیشنهاد می‌کند از حداقل شواهد کافی برای پاسخ درست استفاده شود، همان شواهد دقیق cite شوند و سپس مدل متوقف شود. اگر شواهد کافی نیست، مدل باید شکاف اطلاعاتی را شفاف بیان کند، نه اینکه با حدس‌زدن پاسخ را کامل جلوه دهد.

این اصل مخصوصاً برای موضوعات حساس، اطلاعات جاری، داده‌های مالی، حقوقی، پزشکی، محصولی یا هر نوع ادعای قابل‌تغییر اهمیت دارد.

۶. شخصیت و سبک همکاری مدل

GPT-5.5 به‌صورت پیش‌فرض سبکی کارآمد، مستقیم و task-oriented دارد. این برای سیستم‌های تولیدی مفید است، چون پاسخ‌ها متمرکز می‌مانند، رفتار مدل راحت‌تر کنترل می‌شود و مدل از حاشیه‌پردازی غیرضروری دور می‌ماند.

برای دستیارهای روبه‌کاربر، پشتیبانی، کوچینگ و محصولات مکالمه‌ای، باید دو چیز جدا تعریف شود: personality و collaboration style. شخصیت تعیین می‌کند مدل چطور به نظر برسد: لحن، گرمی، مستقیم‌بودن، رسمی‌بودن، شوخ‌طبعی، همدلی و میزان پرداخت‌شدگی. سبک همکاری تعیین می‌کند مدل چطور کار کند: چه زمانی سؤال بپرسد، چه زمانی فرض معقول بگیرد، چقدر proactive باشد، چه مقدار زمینه بدهد، چه زمانی کارش را بررسی کند و چطور با ریسک یا عدم قطعیت برخورد کند.

هر دو بخش باید کوتاه باشند. شخصیت نباید جایگزین هدف، معیار موفقیت، قوانین ابزار یا شرط توقف شود. شخصیت تجربه کاربر را شکل می‌دهد؛ سبک همکاری رفتار کاری مدل را شکل می‌دهد.

موضوع	شخصیت / Personality	سبک / Collaboration style همکاری
تمرکز	مدل چطور به نظر برسد	مدل چطور کار کند
نمونه‌ها	لحن، گرمی، رسمی‌بودن، همدلی، شوخ‌طبعی	زمان سؤال، فرض معقول، بررسی کار، proactive بودن، مدیریت ریسک

۷. Preamble و به‌روزرسانی‌های میانی

در کارهای چندمرحله‌ای، ابزارمحور یا طولانی، بهتر است قبل از tool call یا شروع کار سنگین یک پیام کوتاه قابل‌مشاهده به کاربر داده شود. این پیام باید درخواست را تأیید کند و قدم اول را بگوید. چنین پیامی به کاربر کمک می‌کند بفهمد مدل چه کاری را شروع کرده و چرا ممکن است کمی زمان ببرد.

راهنما پیشنهاد می‌کند این preamble فقط یک یا دو جمله باشد و به جای توضیح جزئیات فنی، روی فهم درخواست و قدم اول تمرکز کند. برای coding agentهایی که phase جداگانه دارند، می‌توان صریح‌تر گفت پیام میانی باید در phase مناسب منتشر شود.

به‌روزرسانی‌های میانی نباید شبیه گزارش‌نویسی خشک و تکراری باشند. بهتر است فقط در نقاط مهم، هنگام تغییر برنامه یا کشف نکته‌ای مهم، پیام کوتاه و باارزش داده شود.

۸. فرمت خروجی و text.verbosity

GPT-5.5 روی ساختار و فرمت خروجی steerable است. یعنی می‌توان با دستورهای واضح، سطح جزئیات، قالب، بخش‌بندی، طول پاسخ و سبک نمایش را کنترل کرد. مقدار پیش‌فرض text.verbosity در API برابر medium است و وقتی پاسخ کوتاه‌تر می‌خواهید، می‌توان از low استفاده کرد.

راهنما پیشنهاد می‌کند فرمت سنگین فقط زمانی استفاده شود که فهم یا تجربه محصول را بهتر کند. تیتر، بولت، bold، جدول و فهرست شماره‌دار برای مقایسه، رتبه‌بندی، گزارش و پاسخ‌های طولانی مفیدند، اما در مکالمه‌های ساده ممکن است پاسخ را شلوغ کنند.

برای ویرایش، بازنویسی، خلاصه‌سازی یا پیام‌های روبه‌مشتری، باید مشخص شود چه چیزهایی حفظ شوند: ساختار، طول، ژانر، هدف و محتوای اصلی. سپس از مدل خواسته شود وضوح، جریان و دقت را بهتر کند، بدون افزودن ادعای جدید یا تغییر هدف متن.

۹. بودجه بازیابی اطلاعات و citation

برای پاسخ‌های مبتنی بر منبع، باید در پرامپت مشخص شود چه ادعاهایی نیاز به citation دارند، چه چیزی شواهد کافی محسوب می‌شود و اگر شواهد پیدا نشد مدل چه کند. این همان مفهوم retrieval budget است: یعنی قانون توقف و ادامه برای جست‌وجو و بازیابی اطلاعات.

الگوی پیشنهادی این است که برای پرسش‌های عادی، با یک جست‌وجوی broad اما کوتاه و دقیق شروع شود. اگر نتایج برتر درخواست اصلی را پوشش دادند، مدل باید از همان‌ها پاسخ دهد و جست‌وجوی اضافه نکند. جست‌وجوی بعدی فقط وقتی لازم است که سؤال اصلی بی‌پاسخ مانده، یک fact مهم کم باشد، کاربر پوشش جامع خواسته باشد، سند/URL/ایمیل/رکورد خاصی باید خوانده شود، یا پاسخ بدون جست‌وجوی بیشتر شامل ادعای مهم بی‌پشتوانه می‌شود.

جست‌وجوی اضافه برای بهتر کردن جمله‌بندی، افزودن مثال غیرضروری یا cite کردن جزئیات کم‌اهمیت توصیه نمی‌شود.

۱۰. guardrail برای نوشتار خلاقانه

در کارهای خلاقانه یا تولیدی مثل اسلاید، متن معرفی، خلاصه برای اشتراک‌گذاری، talk track، پیام رهبری یا روایت محصولی، باید بین factهای منبع‌دار و wording خلاقانه تمایز گذاشت.

مدل می‌تواند متن را روان، جذاب و متقاعدکننده بنویسد؛ اما نباید برای قوی‌تر کردن متن، نام مشتری، عدد، متریک، وضعیت roadmap، نتیجه مشتری، قابلیت محصول یا ادعای رقابتی را از خودش بسازد. ادعاهای مشخص درباره محصول، مشتری، تاریخ، قابلیت، roadmap، متریک و رقبا باید از منابع بازیابی‌شده یا داده‌های ارائه‌شده بیایند و cite شوند.

اگر پشتیبانی منبعی کافی وجود ندارد، مدل باید draft عمومی مفید با placeholder یا فرض‌های برچسب‌خورده بنویسد، نه اینکه ادعای دقیق جعل کند.

۱۱. فرانت‌اند، طراحی بصری و سلیقه UI

برای کارهای frontend، راهنما توصیه می‌کند به مدل زمینه محصول و کاربر داده شود و از آن خواسته شود با design system، کاربردپذیری صفحه اول، کنترل‌های آشنا، stateهای مورد انتظار، رفتار responsive و کیفیت واقعی UI هماهنگ باشد.

همچنین باید از پیش‌فرض‌های کلیشه‌ای خروجی‌های تولیدی پرهیز کرد؛ مثل heroهای عمومی، کارت‌های تودرتو، گرادیان‌های تزئینی بی‌دلیل، متن‌های آموزشی آشکار و layoutهای شکننده یا خراب.

قاعده عملی: از مدل نخواهید فقط «یک UI زیبا» بسازد؛ بگویید UI برای چه کاربری، چه کار اصلی، چه صفحه‌ای، چه stateهایی و چه محدودیت‌های محصولی طراحی می‌شود.

۱۲. وادار کردن مدل به بررسی کار خودش

هرجا validation ممکن است، باید از مدل خواست خروجی را بررسی کند. برای coding agentها، validationهای مشخص می‌تواند شامل unit test هدفمند برای رفتار تغییرکرده، lint، type check، build check برای packageهای تحت‌تأثیر یا smoke test کوچک باشد.

اگر validation کامل پرهزینه یا غیرممکن است، مدل باید دلیل را توضیح دهد و بهترین بررسی جایگزین را انجام دهد. برای artifactهای بصری، باید خروجی render شود و از نظر clipping، layout، فاصله‌ها، محتوای جاافتاده و هماهنگی بصری بررسی شود.

برای برنامه‌های مهندسی و planning، traceability مهم است: نیازمندی‌ها، محل پوشش هرکدام، منابع/فایل‌ها/APIها/سیستم‌های دخیل، data flow یا validation commands، state transition، رفتار در شکست، ملاحظات privacy/security و سؤال‌های باز مهم باید مشخص باشند.

۱۳. مدیریت phase در workflowهای طولانی

در Responses API و workflowهای tool-heavy، مقدار phase می‌تواند پیام‌های میانی را از پاسخ نهایی جدا کند. اگر از previous_response_id استفاده شود، API وضعیت قبلی assistant را خودکار

حفظ می‌کند. اما اگر اپلیکیشن خروجی‌های قبلی assistant را دستی replay کند، باید phase اصلی هر assistant item دقیقاً حفظ شود.

قاعده عملی این است: هنگام replay دستی assistant itemها، phaseها را تغییر ندهید. برای updateهای میانی قابل‌نمایش از phase مناسب مثل commentary استفاده کنید، برای پاسخ کامل‌شده از final_answer استفاده کنید، و به پیام‌های user مقدار phase اضافه نکنید.

این نکته جلوی یک failure mode رایج را می‌گیرد: اینکه مدل یا سیستم، update میانی را با پاسخ نهایی اشتباه بگیرد یا context مرحله‌های قبلی را بد بازسازی کند.

۱۴. مهاجرت خودکار با Codex

راهنما اشاره می‌کند که Codex می‌تواند با OpenAI Docs Skill تغییرات پیشنهادی برای مهاجرت به GPT-5.5 را روی پروژه اعمال کند. دستور نمونه‌ای که در مستندات آمده چنین است: $ openai-docs migrate this project to gpt-5.5

برای استفاده از این skill در coding agentهای دیگر، باید آن را از مخزن OpenAI skills دریافت کرد. این بخش نشان می‌دهد OpenAI مهاجرت پرامپت‌ها را فقط یک کار متنی نمی‌داند، بلکه آن را بخشی از نگهداری و تکامل سیستم‌های agentic می‌بیند.

با این حال، حتی اگر از Codex استفاده شود، تیم محصول باید prompt جدید را با evalها، سناریوهای واقعی و معیارهای کیفیت خودش تست کند.

۱۵. چک‌لیست سریع طراحی پرامپت برای GPT-5.5

یک پرامپت خوب برای GPT-5.5 معمولاً این پرسش‌ها را جواب می‌دهد: خروجی خوب دقیقاً چیست؟ چه معیارهایی نشان می‌دهند کار کامل شده؟ کدام محدودیت‌ها واقعاً مهم‌اند؟ کدام ادعاها باید منبع داشته باشند؟ چه زمانی باید ابزار استفاده شود؟ چه زمانی باید سؤال پرسیده شود؟ چه زمانی باید متوقف شد؟ خروجی نهایی چه فرمت و چه سطح جزئیاتی داشته باشد؟

اگر پرامپت شما پر از دستورهای ALWAYS و NEVER است، بررسی کنید کدام‌ها واقعاً invariant هستند و کدام‌ها بهتر است به decision rule تبدیل شوند. اگر پرامپت شما مسیر را بیش از حد دیکته می‌کند، آن را به success criteria، outcome و stop rules تبدیل کنید.

اگر خروجی با منبع، کد، فایل، اسلاید، UI ،PDF یا اقدام واقعی سروکار دارد، validation را صریح کنید.

اگر پاسخ کاربرمحور است، شخصیت و سبک همکاری را کوتاه و جداگانه تعریف کنید.

۱۶. قالب آماده پرامپت جامع برای پروژه‌ها

در بخش بعد یک قالب آماده فارسی ارائه شده است. این قالب را می‌توانید برای دستیارهای محصولی، ابزارهای داخلی، coding agentها، دستیارهای پشتیبانی یا workflowهای پژوهشی تنظیم کنید. مهم است که هر بخش را کوتاه نگه دارید و فقط جزئیاتی را اضافه کنید که واقعاً رفتار مدل را تغییر می‌دهد.

نمونه شخصیت task-focused

```
# Personality

.تو یک همکار توانمند، آرام، قابل‌اعتماد و مستقیم هستی

.فرض کن کاربر صلاحیت دارد و با نیت خوب عمل می‌کند

.با صبر، احترام و کمک عملی پاسخ بده

.وقتی درخواست به‌اندازه کافی روشن است، پیشرفت را به توقف برای سؤال ترجیح بده

.فقط زمانی سؤال بپرس که نبود اطلاعات واقعاً نتیجه را تغییر می‌دهد یا ریسک مهمی ایجاد می‌کند

.سؤال‌ها را محدود و دقیق نگه دار

.مختصر باش اما خشک و بی‌روح نه

.به‌اندازه‌ای زمینه بده که کاربر پاسخ را بفهمد و به آن اعتماد کند
```

نمونه قانون توقف

```
Resolve the user query in the fewest useful tool loops,

but do not let loop minimization outrank correctness,

fallback evidence, calculations, or required citations.

After each result, ask:
```

```
Can I answer the user's core request now with useful evidence
and citations for factual claims?
If yes, answer.
```

نمونه retrieval budget

```
Make another retrieval call only when:
– The top results do not answer the core question.
– A required fact, parameter, owner, date, ID, or source is missing.
– The user asked for exhaustive coverage, comparison, or a comprehensive list.
– A specific document, URL, email, record, or code artifact must be read.
– The answer would otherwise contain an important unsupported factual claim.

Do not search again just to improve phrasing or add nonessential examples.
```

قالب جامع پیشنهادی

```
Role:
تو یک دستیار متخصص برای [زمینه/محصول/تیم هستی].
وظیفه تو کمک به کاربر برای رسیدن به [نتیجه مشخص] با پاسخ‌های دقیق، قابل‌اعتماد و کاربردی است.

# Personality
لحن تو حرفه‌ای، روشن، صمیمی و مستقیم است.
مختصر بنویس اما اطلاعات لازم برای اعتماد به پاسخ را حذف نکن.

# Goal
درخواست کاربر را تا حد ممکن کامل و عملی حل کن.
```

روی نتیجه نهایی تمرکز کن، نه توضیح فرایندهای غیرضروری.

Success criteria

– مسئله اصلی کاربر پاسخ داده شود.

– محدودیت‌ها رعایت شوند.

– ادعاهای مهم با منبع یا شواهد پشتیبانی شوند.

– اگر داده‌ای کم است، فقط کوچک‌ترین سؤال ضروری پرسیده شود.

Retrieval and citations

با حداقل جست‌وجوی کافی شروع کن.

اگر نتایج برای پاسخ اصلی کافی بود، جست‌وجوی اضافه انجام نده.

اگر شواهد کافی نیست، شفاف بگو چه چیزی پیدا شد و چه چیزی کافی نبود.

Validation

اگر خروجی قابل بررسی است، آن را بررسی کن.

اگر validation ممکن نیست، دلیل و بهترین جایگزین را بگو.

Stop rules

وقتی پاسخ اصلی با شواهد و جزئیات کافی آماده شد، متوقف شو.

جمع‌بندی نهایی

راهنمای GPT-5.5 در یک جمله می‌گوید: کمتر مسیر را کنترل کن، بیشتر مقصد را تعریف کن. به‌جای prompt stackهای طولانی و فرایندمحور، باید پرامپت‌هایی ساخت که outcome، معیار موفقیت، محدودیت‌ها، شواهد، فرمت خروجی، قوانین توقف و validation را روشن کنند. برای محصولات واقعی هم باید شخصیت، سبک همکاری، preamble، citation rules، retrieval budget و phase handling را متناسب با تجربه کاربر تنظیم کرد.

منابع

OpenAI Developers – Prompt guidance for GPT–5.5:

https://developers.openai.com/api/docs/guides/prompt-guidance?model=gpt-5.5