



خلاصه دوره‌های هوش مصنوعی گوگل – یادداشت مطالعه به فارسی

ساخت عامل با Agent Development Kit

Develop Agents with ADK / Build AI Agents with Google (GEAR)

ارائه‌دهنده: Google Cloud / Google for Developers

۸ فعالیت شامل آزمایشگاه مهارتی

سطح: پیشرفته – نیازمند پایتون

فایل ۲۰ از ۲۲ – تهیه شده برای تدریس

موضوع: ساخت، آزمودن و استقرار عامل با چارچوب ADK **تعداد درس:** ۸ فعالیت **سطح:** پیشرفته
پیش‌نیاز: پایتون + آشنایی با مفاهیم عامل **خروجی:** نشان مهارتی

خلاصه کلی دوره

اگر دوره قبلی مفهوم عامل را می‌سازد، این دوره **دست به کد** می‌شود. محورش **ADK** است: چارچوب متن‌باز گوگل برای ساخت عاملی که می‌تواند برنامه بریزد، استدلال کند و وظیفه را اجرا کند. مسیر با یک آزمایشگاه مهارتی پایان می‌یابد و نشان می‌دهد.

این دوره بخشی از برنامه **GEAR (Gemini Enterprise Agent Ready)** است – مسیری که گوگل برای آماده کردن توسعه‌دهندگان برای اکوسیستم عامل سازمانی طراحی کرده.

نکات مهم هر بخش

۱. آشنایی با ADK

- چارچوب **متن‌باز و مستقل از مدل**: می‌توانید با Gemini کار کنید یا مدل دیگری.
- فلسفه طراحی: عامل مثل نرم‌افزار معمولی نوشته، نسخه‌بندی، تست و مستقر شود.
- نصب، ساختار پروژه، و ساخت یک عامل حداقلی در چند خط کد.

۲. اجزای یک عامل در ADK

- **Agent** – تعریف هدف، دستورالعمل و مدل.
- **Tools** – توابعی که عامل می‌تواند صدا بزند؛ توصیف دقیق هر ابزار حیاتی است چون مدل از روی همان تصمیم می‌گیرد کی صدایش بزند.
- **Session / State** – نگه‌داشتن وضعیت گفت‌وگو و کار.
- **Memory** – حافظه بلندمدت فراتر از یک نشست.
- **Runner** – اجراکننده حلقه عامل.

۳. ابزارها و اتصال به دنیای بیرون

- ابزار سفارشی (تابع پایتون ساده).
- ابزارهای آماده: جست‌وجو، اجرای کد، بازیابی از منبع داده.

- اتصال از طریق MCP به سرویس‌های بیرونی.
- **قاعده طلایی:** توصیف ابزار را مثل مستندات API برای یک همکار تازه‌وارد بنویسید – مبهم بنویسید، عامل اشتباه صدا می‌زند.

۴. سامانه‌های چندعاملی

- تقسیم کار بین چند عامل تخصصی به جای یک عامل گول‌پیکر.
- الگوهای ارکستراسیون: **ترتیبی، موازی، و حلقه‌ای.**
- عامل **هماهنگ‌کننده** که وظایف را به عامل‌های زیردست واگذار می‌کند.
- پروتکل **A2A** برای ارتباط عامل‌هایی که در سامانه‌های مختلف اجرا می‌شوند.

۵. ارزیابی و آزمون عامل

- چرا تست عامل سخت‌تر از تست تابع است: خروجی قطعی نیست.
- ساخت **مجموعه ارزیابی** (eval set): سناریوهای نمونه با نتیجه مورد انتظار.
- سنجش هم مسیر (آیا ابزار درست را صدا زد؟) و هم نتیجه نهایی.
- ردگیری (tracing) برای فهمیدن اینکه عامل در کدام گام منحرف شد.

۶. ایمنی و مرزها

- محدود کردن دسترسی ابزارها بر اساس **کمترین امتیاز لازم.**
- گذاشتن **نقطه تأیید انسانی** برای اقدامات پرریسک.
- محافظت در برابر **تزریق پرامپت** از طریق محتوای بیرونی.
- سقف گذاشتن روی تعداد گام‌ها و هزینه برای جلوگیری از حلقه بی‌پایان.

۷. استقرار

- اجرای محلی برای توسعه و اشکال‌زدایی.
- استقرار روی **Vertex AI Agent Engine** یا **Cloud Run**.
- پایش، لاگ‌برداری و مدیریت هزینه در محیط عملیاتی.

۸. آزمایشگاه مهارتی

- ساخت یک عامل کامل از صفر و دریافت نشان مهارتی.

جمع‌بندی سریع برای مرور

- ADK عامل را به یک **مصنوع نرم‌افزاری** تبدیل می‌کند: کد، تست، نسخه، استقرار.
- سه چیزی که کیفیت عامل را تعیین می‌کند: **توصیف ابزار**، **دستورالعمل روشن**، **مجموعه ارزیابی**.
- به چندعاملی زود نروید؛ اول یک عامل تک را درست کنید.

پرتکرارترین اشتباه

توصیف ابزار مبهم. اگر عامل ابزار اشتباه را صدا می‌زند، نود درصد مواقع مشکل از مدل نیست، از متن توصیف ابزار است.

منابع

- <https://www.skills.google/paths/3545>
- <https://developers.google.com/program/gear>
- <https://cloud.google.com/run/docs/ai/build-and-deploy-ai-agents/adk>