



# دنیای صفر و یک

00010100

شماره بیستم / آذرماه ۱۴۰۴

نشریه تخصصی انجمن علمی مهندسی کامپیوتر دانشکده فاریابی دانشگاه تهران

در این شماره خواهید خواند:

تاریخچه ماندگار بازی‌ها

در باب بهترین بازی سال

مصاحبه با دکتر عادلہ بیطرفان

نگاهی بر اخبار دنیای صفر و یک

نقد و بررسی فیلم **Transcendence**

هفت خان توسعه بک‌اند، خان صفرم و خان اول

**CAP THEOREM**: وقتی سیستم‌های توزیع شده باید انتخاب کنند



# دنیای صفر و یک

## شناسنامه

دانشگاه صادرکننده مجوز:

دانشگاه تهران

صاحب امتیاز:

انجمن علمی مهندسی کامپیوتر دانشکدگان فارابی دانشگاه تهران

مدیر مسئول:

زهرا کمالیان

سر دبیر:

سبحان ولی زاده درخشان

استاد مشاور علمی:

دکتر حسین آقابابا

کارشناس نشریات:

مریم شریفی دانا

زمینه انتشار:

علمی، تخصصی

تاریخ / ترتیب انتشار:

۱۴۰۴/۹/۳۰ - دو ماهنامه

طراح جلد و صفحه آرا:

زهرا پورفرزین، نرگس رضاییان

ویراستار ادبی:

مشکات سلیمی نمین

هیئت تحریریه:

مهرداد پوریوسف، علیرضا کامرو، عرفان رضانی مقدم، احمد محمدی،

فاطمه نصیری، زهرا مهدی خانی، سیده الهه دهقان، رضا قمی الاصل،


ماهان قدیم خانی، سید علی فقیه موسوی، مهدی امامی

نویسندگان:

رضا قمی الاصل، سید علی هدایتی، سید حسین موسوی فرد،

حسین نوری، سیده فاطمه فرحناک فر، محمدجواد پاپی زاده

راه ارتباطی:

 donyayesefroyek.ut@gmail.com

 <http://cesasj.ut.ac.ir>   @donyayesefroyek



# فهرست

## سخن آغازین

۱..... سخن سردبیر

## کیسول زمان

۲..... صفر تا صد دنیای صفر و یک

## یک کلام حرف حساب

۷..... مصاحبه با دکتر عادلہ بیطرفان

## روز روز کاری رایانه

۱۰..... تاریخچه ماندگار بازی‌ها

## تخصص

۱۲..... CAP THEOREM

۱۴..... Edge Computing

۱۷..... انقلاب کیوبیت‌ها

۲۰..... معماری زبان Go

۲۳..... درآمدزایی از داده

## کارتریج

۲۷..... نقد فیلم Transcendence

۲۹..... در باب بهترین بازی سال

## درس دیجیتال

۳۴..... هفت خان توسعه بک‌اند، خان صفرم و خان اول

۳۸..... جعبه ابزار مهندسی

## پچ نوت

۴۰..... نگاهی بر اخبار دنیای صفر و یک

# سخن آغازین

به نام خداوند قلم

سلام به شما رفقای دنیای صفر و یک!

اگر بخوام صادقانه از شروع این دوره بگویم، باید اعتراف کنم که فکر می‌کردیم ماجرای دوره ششم خیلی ساده‌تر باشد؛ اما مثل همیشه، وقتی کار شروع شد، فهمیدیم پشت هر شماره از نشریه دنیایی از هماهنگی، تلاش و وقت گذاشتن هست که فقط وقتی وسط کار می‌افتی، معنیش رو کامل می‌فهمی.

یه انگیزه کوچیک، چندتا آدم که دلشون می‌خواست کاری رو دوباره زنده کنن، و یه نشریه که سال‌ها قبل‌تر با تلاش بچه‌های قبلی ساخته شده بود و حالا رسیده بود دست ما.

برای ما، نشریه فقط چند صفحه مطلب نیست؛

بیشتر شبیه یه دفتر خاطرات جمعیه.

دفتر روزهایی که هرکدومون یه چیزی یاد گرفتیم، یه تجربه داشتیم، یه ایده نوشتیم یا حتی فقط حس کردیم «کاش یکی اینا رو بخونه».

شماره‌های قبلی هم همین‌طور شکل گرفتن؛

با شوق و انرژی آدم‌هایی که قبل از ما بودن و باعث شدن مسیر برای ما روشن‌تر بشه.

حالا نوبت ما بود که این مسیر رو ادامه بدیم.

دوره ششم از دل همین فکرها شروع شد؛ وسط کلاس‌ها، پروژه‌ها، دغدغه‌های دانشگاهی و همه شلوغی‌هایی که می‌دونید چقدر وقت‌گیرن.

ولی کم‌کم کنار هم جمع شدیم، تیممون بزرگ‌تر شد، ایده‌ها رو ریختیم وسط، بحث کردیم، نوشتیم، پاک کردیم و دوباره ساختیم...

و نتیجه شد همین شماره‌ای که الان دست شماست.

امید ما اینه که دنیای صفر و یک برای شما فقط یه نشریه نباشه؛ یه فضای راحت برای یاد گرفتن، خوندن چیزهای تازه، آشنا شدن با تجربه‌های واقعی و حتی خندیدن به لحظه‌های مشترک دانشجویی.

این شماره ادامه‌ی راه بچه‌های قبل از ماست، اما رنگ و حال‌وهوای خودمون رو هم داره؛

ترکیبی از تجربه، انرژی، اشتباه، تلاش و رفاقت.

اگر نظری، انتقادی یا پیشنهادی برای بهتر شدن نشریه دارید، با روی باز می‌شنویم؛

این نشریه تا وقتی ارزش داره که شما بخونیدش، از لذت ببرید، کمک کنید بهتر بشه و در شکل‌دادنش سهم داشته باشید.

ما هم قول می‌دیم هر شماره یه قدم جلوتر باشیم.

امیدوارم شماره بیستم براتون جذاب، مفید و دوست‌داشتنی باشه.

مرسی که کنار ما هستید.

ارادتمند شما

سبحان ولی‌زاده درخشان



**سید علی فقیه موسوی**  
 دانشجوی مهندسی کامپیوتر  
 دانشکده فارابی دانشگاه تهران  
 Faghih.mousavi@ut.ac.ir



## صفر تا صد دنیای صفر و یک

۱۰ دقیقه

### شاید مقدمه

شاید آن روزها که مشغول نوشتن «[قصه ما به سر رسید](#)» در نشریه هجدهم بودم، فکرش را هم نمی‌کردم بار دیگر شرایط به سمتی برود که قلم به دست بگیرم و برای «دنیای صفر و یک» شروع به نوشتن کنم. حال اما با شروع نهمین دوره انجمن علمی مهندسی کامپیوتر دانشکده فارابی دانشگاه تهران، دنیای صفر و یک نیز جان تازه‌ای گرفته و دوباره چرخ‌هایش به حرکت درآمده است. امید است که این انرژی و شعف تا انتهای این دوره نشریه باقی بماند و دوره ششم نشریه منتهی به نتایجی باشد که در شأن نام دنیای صفر و یک است.

و اما بعد، عدد بیست در فرهنگ فارسی همواره دارای معانی و مفاهیم گوناگون بوده است. به نوعی شاید نشریه شماره بیستم را بتوان ایستگاهی دانست که می‌توان در آن توقف کرد و به نوزده شماره قبلی

نگاه انداخت؛ نوزده شماره‌ای که شاید هر کدام در دوره‌های مختلف و با سختی‌های متفاوت به مرحله انتشار رسیده‌اند. از آنجایی که در جریان جزئی‌ترین اتفاقات هفت شماره از این نوزده شماره بوده‌ام، قرعه مرور این نوزده شماره به نام من افتاد. امیدوارم که در این مسیر موردی را از قلم نیندازم و حق مطلب به نحو احسن ادا شود.

در نهایت لازم می‌دانم متذکر شوم از آنجایی که من سهمی در دوره‌های اول، دوم و پنجم نداشتم، اطلاعاتم از این دوره‌ها مربوط است به مطالعه این شماره‌های نشریه و شنیده‌ها از سایر دوستانی که مستقیم یا غیرمستقیم در این دوره‌ها نقش ایفا کرده‌اند.

### دنیای صفر و یک

نشریه دانشجویی دنیای صفر و یک، نشریه تخصصی حوزه کامپیوتر به صاحب امتیازی انجمن علمی مهندسی کامپیوتر

دانشکده فارابی دانشگاه تهران؛ نشریه‌ای که در سال ۱۳۹۶ تاسیس شد و اولین شماره‌اش در اسفند ماه همان سال به انتشار رسید.

از آن سال تا امروز، دنیای صفر و یک به عنوان یکی از فعال‌ترین نشریات دانشجویی، نقش مهمی در ارتقای سطح علمی و ایجاد بستری برای تبادل نظر و ارائه ایده‌های نو برای دانشجویان داشته. این نشریه با رویکردی دانشجوی محور و تخصصی، به صورت دوره‌ای به انتشار مقالات، مصاحبه‌ها، گزارش‌ها و مطالب متنوع در حوزه‌های مختلف کامپیوتر پرداخته.

دنیای صفر و یک در طی سال‌های گذشته درگیر تغییر و تحولات متعددی شده، با انجمن‌های مختلف سر و کار داشته و سکان هدایت آن در طول این سال‌ها شش بار جابه‌جا شده است. در دل این جابه‌جایی‌ها، محتوا، قالب،

# کیس‌سول زمان

## شناسنامه نشریات دوره اول:

اول تا نهم	شمارگان
علی داوودآبادی فراهانی	مدیرمسئول
محمد خشنود	سردبیر
علی داوودآبادی فراهانی(۴ شماره)، محمد خشنود (۵ شماره)، امیرمسعود جوزدانی(۲ شماره)، محمد مهدی هاشمی(۲ شماره)، احمد قمی(۱ شماره)	گرافیک و صفحه‌آرایی
نامشخص	توالی انتشار
اسفند ۱۳۹۶ اردیبهشت ۱۳۹۷ مهر ۱۳۹۷ دی ۱۳۹۷ خرداد ۱۳۹۸ مهر ۱۳۹۸ آذر ۱۳۹۸ بهمن ۱۳۹۸ خرداد ۱۳۹۹	تاریخ های انتشار

اولین باری را که برای کارهای نشریه پیش خانم شریفی (مسئول نشریات دانشکدگان) رفتم، به‌خوبی به یاد می‌آورم. آن روزها ما در انجمن ششم مشغول تدارک و آماده‌سازی تیم دوره سوم نشریه بودیم. تا نام دنیای صفر و یک مطرح شد، ایشان شروع به تعریف از نشریه در دوره اول کرد، تعریفی که کوتاه هم نبودند، نام علی داوودآبادی فراهانی از زبانش نمی‌افتاد و مکرر تاکید بر این داشت که از ایشان کمک بگیریم. آن‌قدر روی نشریات حساس بود که حتی شماره‌های اول تا نهم را در اختیار ما قرار نداد و می‌گفت آن‌ها را خودش نگه داشته، وقتی مسئول نشریات دانشگاه تا این حد نسبت به یک نشریه در دوره‌ای خاص حساس باشد، یعنی آن نشریه در آن دوره کار خود را به‌خوبی انجام داده است.

دوره‌انتشار و موارد این‌چنینی مکرراً تغییر کرده، اما این تغییرات نه تنها مانع حرکت این نشریه نشده، بلکه این نشریه را در مسیر خود نگه داشته و به سمت افتخاراتی شایسته نیز هدایت کرده.



## دوره اول: دنیای صفر و یک متولد می‌شود...

برخلاف سایر نشریات انجمن‌ها که چند سالی بعد از تأسیس انجمن متولد می‌شوند، دنیای صفر و یک همان سال اول تأسیس انجمن و در دل اولین دوره انجمن علمی مهندسی کامپیوتر دانشکدگان فراهانی متولد شد و اولین شماره‌اش در اسفند ماه همان سال به انتشار رسید.



این نشریه به مدیرمسئولی علی داوودآبادی فراهانی و سردبیری محمد خشنود فعالیت خودش را آغاز کرد. به شناسنامه نشریه در چند شماره اول که نگاه می‌کنم این دو نام را در بخش‌ها و عناوین مختلف مکرر می‌بینم، این موضوع نشان دهنده این است که این دو نفر سهم بسزایی در انتشار نشریه‌های دوره اول داشته‌اند. ترکیب این دو نفر آن‌قدر موفق بوده که عمر همراهی آن‌ها با دنیای صفر و یک حدود دو سال و سه ماه به طول انجامیده و با همراهی هم ۹ شماره از دنیای صفر و یک را به مرحله انتشار رسانده‌اند.



## دوره دوم: ترتیب، ساختار و مقداری چیز دیگر...

تقریباً یک سال پس از انتشار نهمین شماره دنیای صفر و یک، شماره دهم این نشریه به مرحله انتشار رسید. انتشار شماره دهم را به نوعی می‌توان آغاز دوره دوم مدیریت نشریه دانست؛ محمدعلی آصف به عنوان مدیرمسئول و مهران آدووند به عنوان سردبیر وظیفه هدایت نشریه در دوره دوم را به عهده گرفته بودند و این دوره تحت نظر چهارمین دوره انجمن علمی و در شرایط سخت کرونایی مسیر خودش را آغاز کرده بود.

با توجه به فاصله‌ای که افتاده بود، موتور نشریه نیاز به نیروی مضاعف داشت تا به حرکت دربیاید. این نیرو از سمت تیم نشریه به خوبی



هرچند دنیای صفر و یک در دوره اول توالی انتشار مشخصی نداشت و از لحاظ فرمت محتوایی و قالب بصری همواره دستخوش تغییرات مختلفی شده بود، اما در این بین حفظ ترکیب اصلی نشریه و تلاش برای رشد و بقای آن، آن هم در طی همکاری با چند انجمن و حتی در دوران کرونا، نکته‌ای قابل توجه است که نمی‌توان از آن بی‌تفاوت گذشت.

## فعالین نشریه در دوره اول:

علی داوودآبادی فراهانی، محمد خشنود، پیام رنجبر، محمد مهدی هاشمی، مجتبی یزدان‌پناه، امیرمسعود جوزدانی، امیرسالار درویش‌پور، حمید فضلی، سیدعلی آل‌طه، امیرحسین قبادی، امیرعلی گرجی و محمد رشیدخان



آن طرح اولیه، هرچند در ابتدا زمان زیادی از من گرفت، اما زمینه‌ای شد برای ورود حرفه‌ای من به دنیای گرافیک و صفحه‌آرایی. به‌خوبی یاد دارم که آن روزهای اول با چه مشقتی این کار را انجام می‌دادم، اما رفته‌رفته هم دست من گرم‌تر شد و هم نتیجه جافتاده‌تر. همان طراحی اولیه باعث شد گرافیک و صفحه‌آرایی نشریه به دست من بیفتد؛ موضوعی که در دوره چهارم نشریه نیز ادامه‌دار بود. افزون بر آن، نشان فعلی نشریه نیز حاصل طراحی من در همان روزهای ابتدایی دوره سوم است که علیرغم تغییر قالب، همچنان به عنوان نشان رسمی دنیای صفر و یک مورد استفاده قرار دارد.



از دل همان جلسه، دنیای صفر و یک ساختاری منظم و اساسی به خود گرفت. بخش‌های نشریه در دوره سوم شامل بخش‌های «تخصصی»، «روزی روزگاری رایانه»، «یک چهره از دانشکده»، «کلاکت»، «بازی‌های رایانه‌ای» و «سلامت» می‌شد. این بخش‌ها به همت بچه‌های انجمن و تیم نشریه به مرحله انتشار می‌رسیدند. در این دوره، نشریه به صورت ماهنامه منتشر می‌شد و انتشار آن تا روزهای آخر انجمن ششم، یعنی اردیبهشت‌ماه ۱۴۰۲، ادامه داشت.

## فعالین نشریه در دوره سوم:

محمد آزاد، سید علی فقیه موسوی، عرفان صابری، زینب قدوسی زاده، نیما تهرانی فرشید، ملیکا عربزاده، علی رادمرد، مهرداد صادق پور، سعید رزاقی، مصطفی کوتی، زهرا مرادی و ...



## دوره سوم: تولدی باشکوه‌تر از تولد اول...

یادم می‌آید اواسط پاییز سال ۱۴۰۱ بود؛ انجمن ششم روزهای ابتدایی خود را سپری می‌کرد. محمد آزاد که آن روزها از طرف انجمن مأمور احیای مجدد نشریه شده بود، جلسه‌ای ترتیب داد تا اعضای انجمن ایده‌های خود را مطرح کنند. محمد در آن روزها خود، کوه ایده بود و با انرژی و جدیت بسیار آمده بود تا تحولی اساسی در نشریه رقم بزند. صوت آن جلسه را هنوز هم دارم و وقتی پس از سه سال دوباره به آن گوش می‌دهم، با خود معترف می‌شوم که هیچ‌کس نمی‌توانست نشریه را این‌گونه سرپا کند. انرژی آن روزهای محمد مرا به یاد زمانی می‌انداخت که مشغول چیدن تیم انجمن ششم بودیم؛ خاطرم هست با شناختی که از او داشتم، او را به باقی بچه‌ها پیشنهاد کرده بودم و هنوز دو ماه از آن ماجرا نگذشته بود که خودش را به همه ثابت کرد. محمد آزاد در دوره سوم فعالیت نشریه، به عنوان مدیرمسئول و سردبیر، ساختار نشریه را تنظیم کرد تا به نوعی «احیاگر نشریه» لقب بگیرد.

در آن روزها با مسائل و مشکلات بسیاری روبه‌رو بودیم؛ از یک طرف نشریه‌ای که بیش از یک سال غیرفعال بود و از طرف دیگر انجمنی که برای همه اعضا اولین تجربه جدی فعالیت بود. همین مسائل باعث شد انتشار اولین شماره از نشریه به اوایل اسفندماه سال ۱۴۰۱ برسد. یکی از اساسی‌ترین چالش‌هایی که در دوره سوم با آن مواجه بودیم، بحث گرافیک و صفحه‌آرایی نشریه بود. از تیم آن زمان انجمن، تنها من سابقه کار گرافیکی داشتم، اما تجربه من نیز بسیار محدود و مربوط به یک نرم‌افزار خاص بود. خلاصه آن‌قدر گشتیم و کسی حاضر نشد این مسئولیت را بر عهده بگیرد که به اصرار محمد قرار شد من طرح اولیه‌ای ارائه دهم تا ببینیم کار به کجا می‌رسد.

تأمین شد؛ به گونه‌ای که نه تنها چراغ دنیای صفر و یک دوباره روشن شد، بلکه در این دوره نشریه ساختار بصری یک‌دست‌تری را به خود دید و بیش از پیش شمایل یک نشریه دانشجویی را به خود گرفت. طراحی و صفحه‌آرایی این دو شماره نشریه به عهده محمدعلی آصف و محمدصالح قطبانی بود. همچنین دنیای صفر و یک در دوره دوم با انتشار به صورت فصلی تحت عنوان فصل‌نامه به کار خود ادامه داد.

هرچند عمر این دوره نسبت به دوره اول کوتاه‌تر بود، اما همین دو شماره نشریه هم مطالب گوناگونی در حوزه‌های مختلف اعم از مطالب تخصصی جذاب و متفاوت، معرفی آزمایشگاه‌های دانشگاه، معرفی انجمن علمی و فعالیت‌های آن و ... را در خود جا داده بود.



## فعالین نشریه در دوره دوم:

محمدعلی آصف، مهران آدووند، محمدصالح قطبانی، نسیم آدووند، احسان شیرعلی، دانیال فرهنگ، علی کریمی، عارفه زالی، سیدعلی آل‌طه، علیرضا زینی، صبا حیدری‌دوست، امیررضا پارسی‌زاده، مهدی نقدیلو، عارف برهانی، محمد رشیدخان، محمدحسین شیرازی و ...

## شناسنامه نشریات دوره دوم:

شمارگان	دهم و یازدهم
مدیرمسئول	محمد علی آصف
سردبیر	مهران آدووند
گرافیک و صفحه‌آرایی	محمدصالح قطبانی، محمد علی آصف
توالی انتشار	فصلنامه
تاریخ های انتشار	فروردین ۱۴۰۰ شهریور ۱۴۰۰

## شناسنامه نشریات دوره سوم:

شمارگان	دوازدهم تا چهاردهم
مدیرمسئول	محمد آزاد
سردبیر	محمد آزاد
گرافیک و صفحه‌آرایی	سیدعلی فقیه موسوی، زینب قدوسی زاده
توالی انتشار	ماهنامه
مسئول بخش تخصصی	عرفان صابری
تاریخ‌های انتشار	اسفند ۱۴۰۱ فروردین ۱۴۰۲ اردیبهشت ۱۴۰۲



## دوره چهارم: نیروی تازه نفس وارد می‌شود...

دوره ششم انجمن به کار خود پایان داده بود و نوبت به من رسیده بود تا با شروع دوره هفتم و به همراهی تیمم، مسیر تازه‌ای را برای انجمن و نشریه ترسیم کنم. از مدت‌ها قبل از آغاز انجمن هفتم به فکر بخش‌های مختلف انجمن افتاده بودم و برای آن‌ها ساختاری را در ذهن داشتم. در آغاز، همه چیز به خوبی پیش می‌رفت؛ من دبیر شدم، تیمم به طور کامل وارد انجمن شد و تقسیم مسئولیت‌ها همان‌طور که می‌خواستم رقم خورد. اما با جدا شدن محمد آزاد از نشریه، در انتخاب سردبیر با چالش روبه‌رو شدم. سردبیری که برای نشریه در نظر گرفته بودم به دلایلی نتوانست ما را همراهی کند و ما ماندیم و نشریه‌ای که در ابتدای مسیر تازه خود بدون سردبیر بود.

همراهی بچه‌های انجمن هفتم آن‌قدر بالا بود که از این چالش به خوبی عبور کردیم. من به عنوان مدیرمسئول، وظیفه هدایت تیم نشریه را بر عهده گرفتم؛ سید محمد جعفری برای سردبیری داوطلب شد، باقی بچه‌های انجمن نیز بخش‌های

دیگر نشریه را در دست گرفتند تا تیم نشریه شکل و شمایل خود را پیدا کند. ساختار و قالب کلی نشریه که از دوره قبل باقی مانده بود، تغییرات اندکی پیدا کرد و ما با تزریق نیروی تازه‌نفس، دست به بهبود نشریه در بخش‌های مختلف زدیم. برخلاف دوره قبل، در این دوره افراد زیادی در هر شماره نشریه سهم داشتند و به نوعی نیروی آن‌ها و انگیزه‌شان سرعت پیشرفت را بیشتر می‌کرد و از خستگی مسیر می‌کاست.



اگر بخواهم از حضور بچه‌ها در بخش‌های مختلف بگویم: بخش تخصصی نشریه در این دوره بسیار متنوع‌تر شده بود و تعداد نویسندگان آن افزایش پیدا کرده بود. مسئولیت بخش تخصصی به دستان علی رادمرد افتاد و او با همراهی تجربه عرفان صابری این بخش را مدیریت می‌کردند. سیده فرگل ناظم‌زاده «روزی روزگاری رایانه» را می‌نوشت، «یک کلام حرف حساب» را سعید رزاقی و محمدرضا آرمان‌پور گردآوری می‌کردند، «کلاکت» تیم قبلی خود را حفظ کرده بود و در این بخش زینب قدوسی‌زاده، من و سید محمد جعفری را همراهی می‌کرد، «دیگه چه خبر» توسط نیما کوه‌خضری و تیمش آماده می‌شد و علاوه بر این‌ها، در بخش ویژه نیز ردپای قدیمی‌های انجمن مشهود بود. تنوع موضوعات و مهم‌تر از آن تنوع نویسندگان، از نکاتی بود که دنیای صفر و یک را در این دوره خواندنی‌تر می‌کرد.

از نکات قابل توجه دیگر در این دوره، سایت نشریه بود که به صورت مرتب و با تفکیک مقالات و نویسندگان، تمامی مطالب در آن منتشر می‌شد. همچنین در این دوره، ارتباط دنیای صفر و یک با خارج از دانشکدگان نیز افزایش یافت. دنیای صفر و یک در این دوره به صورت دو ماهنامه منتشر می‌شد و تا آخر دوره هفتم انجمن ادامه داشت. «[قصه ما](#) به سر رسید» در شماره هجدهم نشریه، نشانی از آخرین نفس‌های انجمن هفتم بود.

## فعالین نشریه در دوره چهارم:

سید علی فقیه موسوی، سید محمد جعفری، سیده فرگل ناظم‌زاده، علی رادمرد، زهرا سادات بنی‌هاشمیان، سعید رزاقی، محمدرضا آرمان‌پور، عرفان صابری، زینب قدوسی‌زاده، محمد امین عارفی‌نیا، نیما تهرانی فرشید، یحیی حسینی، پویا اردخانی، نیما کوه‌خضری، نرگس رضاییان، سید امین امام، سید محمدرضا مقدس‌نژاد و ...

## شناسنامه نشریات دوره چهارم:

شمارگان	پانزدهم تا هجدهم
مدیرمسئول	سیدعلی فقیه موسوی
سردبیر	محمد آزاد
گرافیک و صفحه‌آرایی	سیدعلی فقیه موسوی، زهرا سادات بنی‌هاشمیان (۲ شماره)
توالی انتشار	دو ماهنامه
مسئول بخش تخصصی	علی رادمرد
تاریخ‌های انتشار	آبان ۱۴۰۲ دی ۱۴۰۲ اسفند ۱۴۰۲ اردیبهشت ۱۴۰۳



# کیسول زمان

موفق شد عنوان «نشریه برگزیده دانشگاه تهران» را به دست آورد و پس از آن نیز به عنوان «نشریه برگزیده گروه فنی مهندسی در شانزدهمین جشنواره بین‌المللی حرکت» انتخاب شد.

در جشنواره حرکت هفدهم اما شرایط کمی متفاوت‌تر شده بود و رقابت نیز بسیار سخت‌تر از قبل بود. با این حال، با توجه به تغییر آیین‌نامه، امتیاز نشریه در انتخاب انجمن برتر نیز مؤثر بود. دوره چهارم دنیای صفر و یک به عنوان درصد قابل توجهی از فعالیت انجمن، نقش به‌سزایی در انتخاب هفتمین دوره انجمن به عنوان «انجمن برتر کشور در گروه فنی و مهندسی» داشت.

به گواه قدیمی‌ترهای جشنواره حرکت، شاید اگر محدودیت «دو عنوان جایزه برای هر انجمن» وجود نداشت، دنیای صفر و یک می‌توانست بار دیگر در بخش نشریه برتر نیز صاحب عنوان شود. در جشنواره هفدهم، این نشریه سبب کسب بزرگ‌ترین افتخار تاریخ انجمن و دانشکدگان و مهم‌ترین افتخار دانشگاه تهران در آن جشنواره شد.

## فعالین نشریه در دوره پنجم:

مهدی امامی، علی ایومن، زهرا پورفرزین، امیرحسین نصراللهی، ماهان قدیم‌خانی، مهدی بهرزوی راد، امیرعلی عابدی، امیرحسین صفایی، سیده الهه دهقان، علی رادمرد، امیرعلی ذبیحی

## شناسنامه نشریات دوره پنجم:

شمارگان	نوزدهم
مدیرمسئول	مهدی امامی
سردبیر	علی ایومن
گرافیک و صفحه‌آرایی	زهرا پورفرزین امیرحسین نصراللهی ماهان قدیم‌خانی
توالی انتشار	فصلنامه
تاریخ‌های انتشار	خرداد ۱۴۰۴

## افتخارات نشریه:

دنیای صفر و یک در طول این ۸ سال و ۵ دوره گذشته، افتخاراتی قابل توجهی کسب کرده است. این نشریه در دوره سوم



## دوره پنجم: مشکلات غیر منتظره سد راه تیم نشریه...

با پایان انجمن هفتم و شروع انجمن هشتم، تیم نشریه نیز دچار تغییراتی شد؛ مهدی امامی به‌عنوان مدیرمسئول و علی ایومن به‌عنوان سردبیر، هدایت تیم نشریه را در دوره پنجم را بر عهده گرفتند. به دلیل مسائلی که در بخش فرهنگی دانشکدگان پیش آمده بود، انجمن هشتم از همان ابتدا موانعی را پیش روی خود می‌دید که باعث می‌شد برنامه‌ریزی‌ها و ایده‌ها نیز به تعویق بیفتند. به طبع شعله این اتفاقات دامن نشریه را هم گرفت و همین امر در کند شدن سرعت انتشار نشریه مؤثر بود. شماره اولی که قرار بود برای ورودی‌های جدید به انتشار برسد، به نوعی قربانی همین اتفاقات شد. جدای از این، مسائلی نیز در دل انجمن رخ داد که روند آماده‌سازی و انتشار دنیای صفر و یک را نیز کندتر می‌کرد. نشریه شماره نوزدهم، تنها شماره این دوره، پس از یک سال وقفه در خردادماه سال ۱۴۰۴ منتشر شد.

ساختار کلی نشریه در این دوره تفاوت بسیاری با دو دوره قبل پیدا کرده بود. طراحی و صفحه‌آرایی مجدداً انجام شده بود و بخش‌های نشریه نیز دچار تغییراتی شده بودند. در این دوره نشریه شامل مطالبی نظیر: مطالب گوناگون تخصصی، مصاحبه و بررسی اخبار می‌شد.



از نکات مثبت این نشریه می‌توان به این اشاره کرد که از مرزهای دانشگاه فراتر رفته و به سراغ شرکت‌های فعال در استان نیز رفته بود؛ این مسئله در این دوره بیش از پیش به چشم می‌آمد. دنیای صفر و یک در این دوره مجوز خود را به «فصلنامه» تغییر داد که تنها یک شماره‌اش به انتشار رسید.





## زهرا مهدرخانی

دانشجویار مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
mahdikhanizahra19@gmail.com



## مصاحبه با دکتر عادلہ بیطرفان:

از مهندسی کامپیوتر در قم تا هوش مصنوعی در شریف و نوم آلمان

۱۱ دقیقه

دوران، علاقه من به ساخت سیستم‌هایی که می‌توانند مثل انسان فکر کنند، شکل گرفت. همین علاقه، انگیزه‌ی اصلی من برای انتخاب گرایش هوش مصنوعی در مقاطع کارشناسی ارشد و دکتری شد.

از تجربه تحصیل در دانشگاه‌های قم و شریف بگوئید. این انتقال چگونه بود؟

هرچند انتخاب دانشگاه قم مطابق با علاقه‌ام نبود ولی سعی کردم در جایی که هستم بهترین باشم و موفق شدم در میان ورودی‌های ۱۳۸۸ دانشگاه قم رتبه اول را کسب کنم و پایه‌های علمی خود را در دوران کارشناسی تقویت کنم. در مقطع کارشناسی ارشد وارد دانشگاه صنعتی شریف شدم و افتخار این را داشتم تا از راهنمایی‌های دکتر سلیمانی، به عنوان استاد راهنما، بهره‌مند شوم. ورود به شریف تجربه‌ای کاملاً متفاوت و چالش‌برانگیز بود؛ سطح دروس و انتظارات اساتید به مراتب بالاتر و فضا رقابتی‌تر بود. در ابتدا، تطبیق با این فضای جدید برایم کمی سخت بود و احساس می‌کردم وارد دنیای متفاوت شده‌ام، اما به تدریج توانستم خود را با شرایط جدید وفق دهم و همین تجربه نقش مهمی در رشد علمی من داشت.

فرصت مطالعاتی در دانشگاه نوم آلمان چگونه برای شما فراهم شد و این تجربه چه دستاوردهایی داشت؟

من از ابتدای دوره دکتری با پروفیسور نصیر نواب و زمینه کاری او

خاتم دکتر بیطرفان، با تشکر از وقتی که در اختیار ما قرار دادید. لطفاً خودتان را به طور مختصر برای خوانندگان نشریه معرفی کنید.

سلام. من عادلہ بیطرفان هستم. دارای مدرک کارشناسی مهندسی کامپیوتر (نرم افزار) از دانشگاه قم (ورودی ۱۳۸۸)، کارشناسی ارشد (ورودی ۱۳۹۲) و دکتری (ورودی ۱۳۹۷) گرایش هوش مصنوعی از دانشگاه صنعتی شریف. همچنین به مدت سه سال (از سال ۱۴۰۱ تا ۱۴۰۴) دوره فرصت مطالعاتی را به عنوان پژوهشگر در دانشگاه فنی مونیخ (TUM) آلمان گذراندم. حوزه تخصصی من یادگیری عمیق (Deep Learning) است و تمرکز اصلی‌ام روی تحلیل تصاویر پزشکی است.

مسیر تحصیلی شما از مهندسی کامپیوتر در دانشگاه قم شروع شد. چه شد که این رشته را انتخاب کردید و چگونه به سمت هوش مصنوعی گرایش پیدا کردید؟

من ابتدا به مهندسی برق علاقه‌مند بودم، اما به دلیل سکونت در قم و محدودیت رشته‌ها در سال ۱۳۸۸، مجبور شدم بدون شناخت و علاقه قلبی وارد رشته مهندسی کامپیوتر در دانشگاه قم شوم. اما به مرور زمان، در طول تحصیل وقتی با دروسی مثل برنامه‌نویسی، طراحی الگوریتم و هوش مصنوعی آشنا شدم، به این رشته علاقه‌مند شدم و در همین

??? بسیاری از دانشجویان می‌خواهند هرچه سریع‌تر وارد فضای پژوهش و مقاله‌نویسی در هوش مصنوعی شوند. برخی می‌گویند بهتر است مستقیماً با یک پروژه یا مقاله شروع کرد و در حین کار، مفاهیم لازم را یاد گرفت. شما چه دیدگاهی دارید؟ آیا می‌توان از همان ابتدا «یادگیری پروژه‌محور» را پیش گرفت؟

این سؤال بسیار مهمی است. به نظر من، اولویت باید با یادگیری پایه‌ای و عمیق مفاهیم باشد، بعد حرکت به سمت مقاله‌خوانی و پژوهش. دلیلش را با یک مثال توضیح می‌دهم: اگر شما بخواهید فقط روی یک پروژه مثل «تشخیص چهره» متمرکز شوید، تنها با راه‌حل‌ها و چالش‌های همان حوزه آشنا می‌شوید و ذهن شما روی یک مسئله خاص محدود می‌شود. اما اگر ابتدا مفاهیم پایه‌ای یادگیری ماشین و یادگیری عمیق (Deep Learning) را به صورت اصولی و با درک ریاضیاتی یاد بگیرید، چارچوب فکری منسجم و قابل‌تعمیمی در ذهن شما شکل می‌گیرد که می‌تواند در حل مسائل مختلف کمک کند.

وقتی این چارچوب را داشته باشید، آنگاه با مطالعه مقالات در حوزه‌های مختلف (مثلاً پردازش زبان طبیعی (NLP) و یا بینایی کامپیوتر (computer vision)) می‌توانید ارتباط بین موضوع‌ها و مفاهیم مختلف را ببینید. اینجاست که خلاقیت متولد می‌شود: شما می‌توانید از ایده یک مقاله در حوزه‌ای استفاده کنید و با درک پایه‌ای که دارید، آن را به حوزه تخصصی خودتان تعمیم دهید. مثل اتفاقی که برای «ترانسفورمرها (transformers)» افتاد: ایده‌ای که ابتدا برای NLP ارائه شد، اما بعدها انقلاب بزرگی در بینایی کامپیوتر ایجاد کرد.

مقاله‌خوانی یک مهارت حیاتی است، اما اگر پایه‌های علمی شما قوی نباشد، ممکن است از خواندن مقالات دلسرد شوید. پیشنهاد من این است:

**قدم اول:** پایه‌های ریاضی و مفهومی را با کیفیت بالا یاد بگیرید.

**قدم دوم:** پروژه‌های عملی متنوع، هرچند کوچک، انجام دهید تا آنچه در مفاهیم پایه‌ای یاد گرفتید برایتان قابل درک شود و تجربه عملی پیدا کنید.

**قدم سوم:** حالا با یک ذهنیت ساختاریافته، شروع به مطالعه مقالات کنید. ابتدا مقاله‌های مروری (Survey) یا مقالات پایه‌ای با ارجاعات بالا در حوزه مورد نظر خود را بخوانید تا یک دید کلی از ایده‌های پیشنهاد شده در آن حوزه را پیدا کنید، سپس به سراغ مقالات جدید با رویکردهای پیشرفته بروید.

??? برخی فکر می‌کنند با رشد سریع ابزارهای هوش مصنوعی، برخی مهارت‌های انسانی ممکن است کمرنگ شود. نظر شما چیست؟

هوش مصنوعی یک ابزار قدرتمند است، اما جایگزین مهارت‌های اصلی انسانی نمی‌شود. خلاقیت، نوآوری، قدرت تعمیم‌دهی بین حوزه‌های مختلف، کار تیمی و ارتباط مؤثر، مهارت‌هایی هستند که همیشه ارزشمند خواهند بود. یک پژوهشگر موفق کسی است که دانش پایه علمی قوی داشته باشد و بتواند با خلاقیت، ایده‌ها را از یک حوزه به حوزه دیگر منتقل کند (مثل ایده ترانسفورمرها از پردازش زبان طبیعی به بینایی کامپیوتر). علاوه بر این، توانایی یادگیری سریع و انطباق با دانش روز، از مزیت‌های همیشگی انسان است.

آشنا بودم و مقالاتش را دنبال می‌کردم. بعد از چاپ چند مقاله که خیلی مرتبط با زمینه کاری پروفیسور بود، آن‌ها را برای ایشان ارسال کردم و درخواست کردم به عنوان پژوهشگر با گروهشان همکاری کنم. خوشبختانه موافقت کردند و این شد که من فرصت مطالعاتی خود را در دانشگاه TUM آغاز کردم. این تجربه برای من بسیار ارزشمند بود. کار در یک دانشگاه بین‌المللی و پیشرفته مثل دانشگاه TUM باعث شد با موضوعات پژوهشی روز دنیا آشنا شوم، با پژوهشگران بین‌المللی ارتباط برقرار کنم و تجربه کار عملی روی پروژه‌های هوش مصنوعی و پزشکی با کیفیت جهانی را به دست بیاورم.

??? حوزه تخصصی فعلی شما چیست و چرا این حوزه را انتخاب کردید؟

تمرکز اصلی من روی یادگیری عمیق و تحلیل تصاویر پزشکی (Medical Image Analysis) است؛ یعنی تحلیل تصاویر سی‌تی، ام‌آر‌آی و سونوگرافی برای تشخیص بیماری‌هایی مثل ناهنجاری‌های قلبی، تومورهای کبدی یا بیماری‌های مغزی مثل آلزایمر، MS و غیره. در ابتدای دوره دکتری، با مطالعه گسترده مقالات متوجه شدم که این حوزه علی‌رغم اهمیت بالا، به دلیل کمبود داده‌های برجسب‌دار، چالش‌های حل‌نشده زیادی دارد. به همین دلیل، فرصت زیادی در این حوزه برای ارائه ایده‌های جدید و راه‌حل‌های نوآورانه در زمینه یادگیری نیمه‌نظارتی و یادگیری ضعیف وجود دارد. علاوه بر این، کار پژوهشی در حوزه «تحلیل تصاویر پزشکی» می‌توانست به طور مستقیم به سلامت و درمان بیماران کمک کند و همین باعث شد انگیزه زیادی پیدا کنم تا مسیر پژوهشی خود را در این زمینه ادامه دهم.

??? با توجه به مسیر پرفراز و نشیب شما، چه توصیه‌ای برای دانشجویان علاقه‌مند به هوش مصنوعی دارید؟

نقشه راه را می‌توان در چند گام خلاصه کرد:

**تقویت پایه‌ها:** ریاضیات (جبر خطی، آمار و احتمالات و بهینه‌سازی) و برنامه‌نویسی (ترجیحاً پایتون) سنگ بنای این رشته هستند.

**یادگیری گام‌به‌گام:** ابتدا مفاهیم پایه یادگیری ماشین (Machine Learning) مانند رگرسیون (regression)، دسته‌بندی، KNN و غیره را به خوبی یاد بگیرید و سپس وارد مباحث شبکه‌های عصبی و یادگیری عمیق شوید.

**کار عملی و پروژه‌محوری:** پس از کسب دانش تئوری، حتماً روی پروژه‌های عملی و واقعی کار کنید. کار با داده‌های واقعی، برخورد با خطاها و رفع باگ‌ها، بهترین درس‌ها را می‌دهد.

**مطالعه مستقل و مقاله‌خوانی:** دنیای هوش مصنوعی خیلی سریع جلو می‌رود و دانشگاه فقط نقطه شروع است. مهارت جستجو و خواندن مقالات علمی را در خود تقویت کنید. این کار شما را با مرزهای دانش و مسائل باز روز دنیا آشنا می‌کند.

**پشتکار و صبر:** هوش مصنوعی پر از چالش‌های فنی است. گاهی مدل درست کار نمی‌کند یا کد اشکال دارد، داده‌ها ممکن است ناقص یا اشتباه باشند، چندین پارامتر برای مدل باید تنظیم شود. برای رفع این چالش‌ها ممکن است روزها وقت صرف کنید. پشتکار و توانایی مدیریت شرایط دشوار، کلید موفقیت است.

سومین چالش بزرگ من مربوط به اپلای برای دکترا بود. یک سال کامل را صرف آماده‌سازی اپلای برای دانشگاه‌های آمریکا کردم و حتی کنکور دکترای داخلی را شرکت نکردم. در نهایت از دانشگاه بوستون آمریکا پذیرش گرفتم، اما به دلیل شرایط سیاسی آن زمان (اوایل دوره ریاست‌جمهوری ترامپ)، ویزا به من و بسیاری از هم‌دوره‌ای‌ها تعلق نگرفت.

توصیه من برای مدیریت این سردرگمی‌ها و شکست‌ها این است:

**مدیریت بحران به جای تسلیم شدن:** وقتی با مانع یا تغییر مسیر ناخواسته‌ای روبرو می‌شوید، به جای تمرکز روی علت شکست یا غرق شدن در ناامیدی، به دنبال «مسیر جایگزین» باشید.



**دیدن فرصت‌های جدید:** گاهی مسیرهایی که مجبور به انتخابشان هستیم، فرصت‌های پنهانی دارند. هر شکست می‌تواند حکمتی داشته باشد و فرصتی برای یادگیری، رشد و قوی‌تر شدن باشد و شما را به سمت هدفی بهتر و روشن‌تر هدایت کند.

**نگاه به زندگی به عنوان یک مسئله بهینه‌سازی:** در ریاضیات، الگوریتم‌ها گاهی در «مینیمم‌های محلی» گیر می‌کنند. برای خروج از آن، باید «گام» بزرگی برداشت یا «نرخ یادگیری» را تغییر داد. در زندگی هم گاهی برای خروج از یک بن‌بست یا سردرگمی، باید شجاعت ریسک کردن، تغییر جهت یا صبر و یادگیری بیشتر (همان افزایش نرخ یادگیری!) را داشته باشیم.

در هوش مصنوعی، الگوریتم‌ها گاهی در «بهینه‌های محلی» گیر می‌کنند و برای خروج از آن، باید نرخ یادگیری را افزایش داد و گام بزرگی برداشت. در زندگی هم گاهی برای عبور از بن‌بست‌ها و یا سردرگمی‌ها، باید شجاعت ریسک کردن، تغییر مسیر یا صرف زمان بیشتر را داشته باشیم تا به «بهینه سراسری» برسیم.

پس به دانشجویان می‌گویم: سردرگمی و حتی شکست بخشی طبیعی از مسیر پیشرفت است. مهم این است که منفعل نباشید، خودتان را به‌طور مستمر تقویت کنید و با هوشمندی به دنبال راه‌حل‌ها و مسیرهای جایگزین باشید. چیزی که الان به نظر شکست می‌آید، می‌تواند در آینده نقطه قوت و فرصت رشد شما باشد.

## ??? به عنوان حرف آخر، چه پیامی برای دانشجویان دارید؟

روی خودتان سرمایه‌گذاری کنید. یادگیری عمیق، صبر و پشتکار همیشه برای شما معجزه خواهند کرد. ممکن است در مسیر با سردرگمی، تغییر مسیر یا شکست‌های مقطعی روبرو شوید، اما به جای ناامیدی، به دنبال راه‌حل و فرصتی مسیر جایگزین باشید. هر شکست می‌تواند حکمتی داشته باشد و فرصتی برای قوی‌تر شدن باشد و حرکت به سمت مقصدی بهتر. دوران دانشجویی فرصت طلایی است؛ به جای تمرکز صرف روی نمره، روی عمق یادگیری و تقویت رزومه پژوهشی خود وقت بگذارید.

**??? بسیاری از دانشجویان در مقاطع مختلف، با انبوهی از انتخاب‌های پیش رو (گرایش، دانشگاه، مسیر شغلی، پژوهش و...) مواجه می‌شوند و این گزینه‌های زیاد می‌تواند منجر به سردرگمی و بلاتکلیفی شود. شما در مسیر خود، از دانشگاه قم به شریف و سپس فرصت مطالعاتی در آلمان، قطعاً با چنین چالش‌هایی روبرو شده‌اید. چگونه با این سردرگمی‌ها دست و پنجه نرم کردید و چه توصیه‌ای برای مدیریت این شرایط دارید؟**

بله، این سردرگمی‌ها را به وضوح تجربه کرده‌ام. چند نمونه را می‌توانم مثال بزنم:

اولین چالش بزرگ من، انتقال از دانشگاه قم به محیط بسیار رقابتی دانشگاه شریف بود. انتظارات بالا بود و باید خیلی سریع خودم را با شرایط جدید وفق می‌دادم. این فشار در ابتدا سخت و پر از استرس بود، اما به مرور تبدیل به محرکی برای رشد و پیشرفت شد. اولین درسی که گرفتم این بود که گاهی لازم است از منطقه امن خود خارج شوید و با شرایط سخت روبرو شوید تا بتوانید رشد کنید.

دومین چالش من مربوط به اوایل دوره ارشد بود. به شدت علاقه‌مند به کار روی دسته‌بندی تصاویر با استفاده از یادگیری عمیق بودم (مثل تشخیص چهره)، اما در آن زمان (حدود سال‌های ۹۳-۹۲) دسترسی به سخت‌افزارهای قدرتمند مثل GPU در ایران بسیار محدود بود. به دلیل همین محدودیت منابع سخت افزاری مجبور شدم موضوع پایان‌نامه ارشدم را تغییر دهم با اینکه ایده پیشنهادی و حتی راه‌حل پیاده‌سازی شده برای تشخیص چهره با استفاده از یادگیری عمیق را داشتیم. در آن لحظه احساس ناامیدی و شکست کردم، چون مدت زیادی را صرف مطالعه در این حوزه کرده بودم، اما ناچار بودم سریع حوزه تحقیقاتی خود را تغییر دهم. اما جالب این بود که همین تجربه اجباری با یک حوزه جدید (domain adaptation)، بعدها در دوران دکترا بسیار به من کمک کرد. توانستم دانش و تجربه‌ای که در این حوزه به دست آورده بودم را با تکنیک‌های یادگیری عمیق ترکیب کنم و مسائل مربوط به domain adaptation را با رویکردهای مدرن حل کنم. بنابراین این تغییر مسیر اجباری، در نهایت حکمتی داشت و توانایی‌های من را گسترش داد.

**سیده الهه دهقان**

دانشجوی مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
edbelaheh85@gmail.com



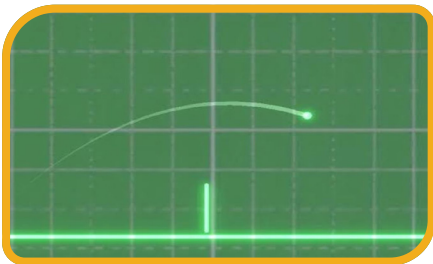
## تاریخچه ماندگار بازی‌ها

1 دقیقه

این ورق‌های سبک با طرح‌های متنوع، ترکیبی از هنر و مسابقه بودند و به تدریج از طریق مسیرهای تجاری راهی غرب شدند. با گذشت زمان، شکل و معنای آن‌ها دستخوش تغییر شد و در اروپا به مجموعه‌ای منظم از نمادها و خال‌ها تبدیل گشت. کارت‌های بازی از همان ابتدا، تلاشی بودند برای پیوند زدن شانس و مهارت در قالبی قابل حمل و ساده.

### نقطه‌ای نورانی در کامپیوتر

قرن‌ها بعد، با ظهور فناوری‌های الکترونیکی، بازی در قالبی کاملاً جدید متولد شد: Tennis for Two (1958). این بازی توسط هیگین‌بوتم بر روی یک اسیلوسکوپ به اجرا درآمد؛ نقطه‌ای نورانی که به دو سوی صفحه رفت و آمد می‌کرد و تنیس دیجیتال را شبیه‌سازی می‌نمود. همین تجربه‌ی ساده، نخستین گام برای ورود بازی‌ها به دنیای دیجیتال و انقلاب بزرگ بعدی در صنعت سرگرمی بود.



### D&D

با گذشت زمان، و در نهایت در سال ۱۹۷۴، بازی‌ها از رقابت‌های ساده فراتر رفتند و به قلمروی جهان‌های خیالی کشیده شد.



در جدال میان نیروهای خیر و شر بود. سنت در ظاهر یک بازی بود، اما در حقیقت تلاشی اولیه برای تبدیل رقابت به یک تجربه‌ی آیینی و معنادار محسوب می‌شد، جایی که سرگرمی با باورهای عمیق فرهنگی آمیخته گشت.

### استخوان‌های کوچک شانس

در شهر سوخته‌ی ایران و سرزمین‌های بین‌النهرین، در حدود ۲۶۰۰ سال پیش از میلاد، انسان‌ها ابزار دیگری برای بازی ساختند: قطعات صیقلی استخوان قوزک حیوانات که بعدها به شکل امروزی تاس تبدیل شدند.



این قطعات نه تنها وسیله‌ای برای سرگرمی، بلکه راهی برای آزمودن شانس و تفسیر بخت بودند. هر پرتاب، حرکتی میان تصادف و انتظار بود و این‌گونه بود که نخستین بازی‌های تاس‌محور به وجود آمدند.

### ورق‌های شرقی

در چین، در زمان دودمان تانگ، هنگامی که کاغذ به ماده‌ای رایج تبدیل شد، نخستین کارت‌های بازی در سده‌ی نهم میلادی شکل گرفتند.



خلق بازی‌ها، با جرعه‌ی کوچکی در دل یکنواختی زندگی‌های تکراری آغاز شد.

اگر تاریخ را ورق بزنیم، به نقطه‌ای می‌رسیم که زندگی انسان‌ها پر از یکنواختی بود؛ روزهایی که کارشان پایان می‌یافت و چیزی برای پر کردن سکوت میان لحظات نداشتند. همان‌جا بود که احساس کسالت و پیدا کردن یک سرگرمی، به جای مشکل، تبدیل به یک جرعه شد؛ جرعه‌ای که انسان را وادار به ساخت چیزی کرد که زمان را برایش خوشایندتر کند.

از همین نیاز ساده، نخستین بازی‌های جهان شکل گرفتند. انسان‌های اولیه با همان ابزار و وسایل اطرافشان -سنگ، چوب، استخوان یا حتی کاغذ- راهی پیدا کردند تا از دل یکنواختی‌ها، رقابت، شانس یا روایت خلق کنند. گویا بشر هر بار که فراغت یافت، دست به اختراع زد؛ و بازی‌ها یکی از اولین ماندگارترین ساخته‌های او بوده‌اند.

در ادامه، قصد داریم با نخستین بازی‌ها از دسته‌های گوناگون و روند پیدایش آن‌ها آشنا شویم.

### از چند مهره تا دنیای ارواح

سنت (Senet)، اولین بازی شناخته‌شده‌ی رومیزی، با چند مهره‌ی ساده بر روی یک تخته‌ی چوبی (حدود ۳۱۰۰ ق.م) آغاز شد. اما برای مصریان باستان، این بازی چیزی فراتر از یک سرگرمی روزمره بود.



آن‌ها حرکت مهره‌ها را بازنمایی از سفر روح در دنیای پس از مرگ می‌دانستند. هر خانه، مرحله‌ای از مسیر معنوی، و هر جابه‌جایی، قدمی

## پرواز در جهانی دیجیتال

با ورود شبیه‌سازها، بازی نه تنها ابزار سرگرمی، بلکه وسیله‌ای برای تجربه‌کردن موقعیت‌های پیچیده شد.



نسخه‌های اولیه‌ی Flight Simulator (1980) تلاش داشتند حس پرواز را - حتی در محدودترین امکانات - بازآفرینی کنند. این تلاش، آغاز دوره‌ای بود که در آن بازی‌ها مرز میان واقعیت و بازآفرینی دیجیتال را کمرنگ ساختند و نقش تازه‌ای در آموزش و تجربه‌سازی یافتند.

## فرماندهی در لحظه

در دهه‌ی ۱۹۹۰، سبک استراتژی به شکل مدرن خود رسید. بازی Dune II الگوی جمع‌آوری منابع، ساخت پایگاه و فرماندهی هم‌زمان به یگان‌ها را تثبیت کرد.



در این سبک، میدان نبرد پویا و زنده بود و هر تصمیم باید در لحظه گرفته می‌شد. این بازی ترکیبی از سرعت، دقت و برنامه‌ریزی را پایه‌گذاری کرد که سال‌ها بعد، در بسیاری از بازی‌های استراتژی، تداوم یافت.

## یک نقشه، صد نفر

در نهایت، مسیر بازی‌ها به سمت رقابت‌های گسترده و محوریت بقا پیش

رفت. (2017) PUBG تجربه‌ای تازه ارائه داد: ورود ده‌ها بازیکن به یک میدان مشترک و تلاش برای ماندن تا آخرین لحظه. این ساختار ساده اما پرتنش، سبک بتل‌رویال را تثبیت کرد و نشان داد چگونه بازی‌ها می‌توانند حس بقا، استراتژی و هیجان را در یک قالب واحد گرد هم آورند.



## پایان یک مسیر، آغاز بازی بعدی...

همان‌طور که دیدیم، بازی‌ها از چند مهره‌ی ساده روی تخته‌ای در مصر باستان آغاز شدند و مسیر طولانی‌ای را طی کردند: از استخوان‌های شانس و کارت‌های کاغذی، تا صفحه‌های نورانی اسیلوسکوپ، جهان‌های خیالی و میدان‌های نبرد دیجیتال. هر بازی، نه فقط یک سرگرمی، بلکه انعکاسی از خلاقیت، رقابت و کنجکاوی انسان بوده است.

تاریخچه‌ی بازی‌ها نشان می‌دهد آنچه روزی تفریحی ساده بود، به‌مرور تبدیل به فرهنگی گسترده، صنعتی پویا و دنیایی بی‌پایان از سرگرمی و داستان‌سرایی شد. و شاید جالب باشد بدانید: هر بار که انسان حوصله‌اش سر می‌رود، صفحه‌ای جدید، سکویی تازه یا جهان دیگری خلق می‌شود. پس می‌توان گفت این مسیر همچنان ادامه دارد و بی‌تردید، بازی بعدی فقط یک قدم جلوتر، منتظر شماست تا کشفش کنید.



Dungeons & Dragons (D&D) نمونه‌ای شاخص بود که برای اولین بار به بازیکنان اجازه داد تا شخصیت‌هایی مستقل بسازند و مسیر داستانی خود را شکل دهند. در این قالب تازه، بازی علاوه بر رقابت، مشارکت، روایت‌پردازی و رویاپردازی را نیز در بر می‌گرفت و گامی بزرگ برای شکل‌گیری بازی‌های نقش‌آفرینی مدرن محسوب می‌شد.

## حرکت روی سکوها

در آغاز دهه‌ی ۱۹۸۰، بازی‌ها به سمت چالش‌های حرکتی و واکنش سریع رفتند. بازی Donkey Kong، با سکوی متحرک، موانع و قهرمانی که باید می‌پرید و بالا می‌رفت، تعریفی تازه از تعامل دیجیتال ارائه کرد و به این صورت، اولین بازی پلتفرمر مدرن به حرکت درآمد. هر مرحله از این بازی، میدان آزمایشی بود برای سنجش مهارت و زمان‌بندی بازیکن، و این سبک به سرعت به یکی از ستون‌های اصلی صنعت بازی‌سازی تبدیل شد.



## شروع جست‌وجوهای پرماجرا

در همان سال‌ها، بازی Adventure (1979) برای کنسول آتاری ۲۶۰۰ (Atari 2600) شکل تازه‌ای از بازی را معرفی کرد: ترکیبی از حل معما، خطر و اکتشاف. بازیکن برای نخستین بار می‌توانست آزادانه مسیر خود را انتخاب کند و در جهانی هرچند ساده، اما غیرخطی حرکت کند. این تجربه، پایه‌گذار ژانر اکشن-ماجراجویی شد؛ سبکی که آزادی و کنجکاوی را در مرکز بازی قرار داد.



## رضا قهرالاسلام

دانشجوی مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
rezaqomyasl@gmail.com



## :CAP THEOREM

## وقتی سیستم‌های توزیع شده باید انتخاب کنند

۱۱ دقیقه

مثال: توی توپیتتر، حتی اگه یکی از دیتاسنترها از کار بیفته، همچنان می‌تونید توپیتت‌ها رو ببینید (شاید کمی قدیمی).

## ۳. Partition Tolerance (تحمل پارتیشن)

یعنی سیستم باید بتونه در شرایطی که ارتباط شبکه بین بعضی از سرورها قطع یا مختل شده (پارتیشن شبکه)، به کار خودش ادامه بده.

این ویژگی در دنیای واقعی اجتناب‌ناپذیره. همیشه این امکان وجود داره که کابل شبکه قطع بشه، سویچ خراب بشه، یا تأخیر شبکه بالا بره. نمی‌تونید فرض کنید که اختلال هرگز اتفاق نمی‌افته.

چرا نمی‌تونیم هر سه رو با هم داشته باشیم؟

فرض کنید یه فروشگاه اینترنتی دارید با دو تا دیتاسنتر - یکی در تهران و یکی در مشهد. موجودی یه کالا روی هر دو سرور «۱۰ عدد» ثبت شده.

حالا یه مشتری در تهران خرید می‌کنه. سرور تهران موجودی رو به «۹» تغییر می‌ده. بعد سعی می‌کنه این تغییر رو به سرور مشهد بفرسته.

تا اینجا مشکلی نیست. اما اگه ارتباط بین تهران و مشهد قطع بشه چی؟

سرور مشهد از تغییر خبر نداره. هنوز فکر می‌کنه موجودی ۱۰ تاست. حالا یه مشتری از مشهد هم همون کالا رو می‌خره.

اینجاست که پارتیشن شبکه رخ داده. سیستم باید یکی از دو انتخاب سخت رو انجام بده:

## انتخاب CP (سازگاری + تحمل پارتیشن)

بگید «تا وقتی مطمئن نیستم، جواب نمی‌دم».

داده درست می‌مونه، ولی سیستم موقتاً از دسترس خارجه. یعنی سرور مشهد درخواست رو رد می‌کنه تا وقتی که بتونه با تهران هماهنگ بشه.

## انتخاب AP (در دسترس بودن + تحمل پارتیشن)

بگید «فعلاً بفروش، بعداً هماهنگ می‌کنیم».

سیستم فعال می‌مونه، ولی داده‌ها ممکنه موقتاً ناسازگار بشن. یعنی ممکنه هر دو سرور کالا رو بفروشن و موجودی منفی بشه.

**نکته مهم:** در عمل، سیستم‌هایی که AP رو انتخاب می‌کنن معمولاً از روش‌هایی مثل صف‌های پیام‌رسانی، قفل‌های توزیع‌شده، یا مکانیزم‌های هماهنگی تدریجی (eventual consistency) استفاده می‌کنن تا بعد از برقراری مجدد ارتباط، تعارض‌ها رو حل کنن و از فروش بیش از حد جلوگیری کنن.

فرض کنید دارید یه اپلیکیشن می‌سازید که قراره میلیون‌ها نفر از شما استفاده کنن. از یه جایی به بعد دیگه یه سرور نمی‌تونه جواب‌گو باشه. پس مجبور می‌شید داده‌هاتون رو بین چند سرور پخش کنید. این کار هم عملکرد رو بهتر می‌کنه، هم اگه یکی از سرورها از کار افتاد، بقیه سرورها هستن و نرم‌افزارتون می‌تونه به کار خودش ادامه بده.

اما اینجا یه سری سوال مهم پیش میاد:

چطور مطمئن بشیم همه‌ی سرورها داده‌ی یکسانی دارن؟

اگه ارتباط بین شون یا مثلاً اینترنت قطع بشه، سرور ها باید چی کار کنن؟

چطور همیشه و سریع به کاربر ها جواب بدیم؟

جواب کوتاه و ناراحت‌کننده اینه که نمی‌تونیم همه‌ی این‌ها رو با هم داشته باشیم!

اینجاست که پای قضیه CAP به میون میاد. مفهومی که می‌گه در سیستم‌های توزیع‌شده، وقتی مشکل شبکه پیش میاد، همیشه باید بین چند هدف مهم، یکی رو فدای دیگری کنیم.

## CAP THEOREM یا قضیه CAP چیه دقیقاً؟

این ایده اولین بار سال ۲۰۰۰، توسط اریک بروئر از دانشگاه برکلی مطرح شد. بعدها به اسم CAP Theorem شناخته شد.

طبق این قضیه، هر سیستم توزیع‌شده در زمان پارتیشن شبکه (اختلال شبکه؛ وقتی ارتباط بین بخشی از سرورها قطع یا مختل می‌شه) فقط می‌تونه دو تا از سه ویژگی زیر رو به‌طور همزمان داشته باشه:

## ۱. Consistency (سازگاری)

یعنی همه‌ی نودها در یک لحظه، داده‌ی یکسانی رو نمایش بدن. اگه یه کاربر داده‌ای رو تغییر داد، همه باید همون تغییر رو بلافاصله ببینن.

مثال: توی سیستم بانکی، وقتی ۵ میلیون از حسابتون برداشت می‌کنید، موجودی در همه‌ی شعب و اپلیکیشن‌ها فوراً به‌روز می‌شه.

## ۲. Availability (در دسترس بودن)

یعنی سیستم همیشه جواب بده - حتی اگه بخشی از اون خراب شده باشه. هر درخواست معتبر باید جوابی دریافت کنه، حتی اگه جدیدترین داده نباشه.

۱. این Consistency با C در ACID متفاوته. در CAP منظور «خواندن آخرین داده نوشته‌شده» هست (linearizability). اما در ACID منظور «رعایت قوانین یکپارچگی داده» مثل کلیدهای خارجی و محدودیت‌هاست.



نکته کلیدی: در شرایط عادی (بدون پارتیشن)، سیستم می‌تونه هر سه ویژگی رو داشته باشه. اما وقتی اختلال شبکه پیش میاد، مجبوره بین Availability و Consistency یکی رو انتخاب کنه.



## دو رویکرد اصلی: BASE و ACID

برای کنار اومدن با محدودیت CAP، دو فلسفه‌ی اصلی در طراحی پایگاه‌داده‌ها شکل گرفته:

### ACID: دقت و نظم در اولویت

پایگاه‌داده‌های سنتی مثل MySQL یا PostgreSQL از مدل ACID استفاده می‌کنن. این مدل چهار ویژگی داره:

**Atomicity (اتمی بودن):** هر تراکنش یا کامل انجام می‌شه یا اصلاً انجام نمی‌شه. برای مثال در انتقال وجه، پول از حساب یک نفر کم و به حساب دیگری اضافه می‌شه. اگر مشکلی پیش بیاد، تراکنش به‌طور کامل لغو می‌شه و این‌طور نیست که پول از حساب مبدأ کم بشه اما به حساب مقصد واریز نشه.

**Consistency (سازگاری منطقی):** داده‌ها همیشه قوانین مشخص رو رعایت می‌کنن (مثل کلیدهای خارجی: مثلاً آگه توی دیتابیس، یه فیلد user\_id داری که به تیبل users اشاره می‌کنه، حتماً باید user با همون id وجود داشته باشه).

**Isolation (جداسازی):** تراکنش‌های هم‌زمان روی هم اثر نمی‌ذارن. مثلاً اگر دو نفر به‌طور هم‌زمان بخوان موجودی حساب رو تغییر بدن، سیستم طوری عمل می‌کنه که انگار هر کدوم جداگانه اجرا شدن و هیچ تداخلی ایجاد نمی‌شه.

**Durability (پایداری):** داده‌ها بعد از ثبت شدن، دیگه از بین نمی‌رن حتی آگه برق دیتاستر به‌طور ناگهانی بره، باز هم داده‌ها به شکل صحیح و سالم سرچاشون هستن.

این نوع سیستم‌ها معمولاً سمت CP هستن. یعنی سازگاری برآشون مهم‌تر از در دسترس بودن آنی هست. آگه شبکه اختلال داشته باشه، ترجیح می‌دن موقتاً جواب نندن تا داده اشتباه بدن.

### BASE: انعطاف و مقیاس بالا

پایگاه‌داده‌های مدرن و توزیع‌شده مثل Cassandra، DynamoDB یا Couchbase معمولاً از مدل BASE استفاده می‌کنن. این مدل سه ویژگی اصلی داره:

**Basically Available (تقریباً در دسترس):** سیستم تقریباً همیشه پاسخ می‌ده.

**Soft State (وضعیت نرم):** وضعیت سیستم ممکنه موقتاً تغییر کنه. مثلاً در یه وب‌سایت فروش آنلاین، وقتی یک کالا به تازگی موجود شده یا موجودی اون تغییر کرده، ممکنه تا مدتی در همه سرورها و کش‌ها یکسان نشون داده نشه و سیستم یه حالت موقت ناپایدار داشته باشه تا همه سرورها به‌روز بشن.

**Eventually Consistent (سازگاری نهایی):** آگه نوشتن جدیدی انجام نشه، در نهایت همه‌جا داده‌ها یکی می‌شن.

این نوع سیستم‌ها معمولاً سمت AP هستن. یعنی حتی در زمان قطعی یا تأخیر، سیستم همچنان پاسخ می‌ده. بعداً داده‌ها رو بین نودها هماهنگ می‌کنه. مثلاً در DynamoDB، داده ممکنه بلافاصله روی همه‌ی سرورها همگام نباشه. اما سیستم تضمین می‌کنه که در نهایت همه به حالت یکسان می‌رسن.

## انتخاب در دنیای واقعی: CP یا AP؟

وقتی داده‌ها برامون خیلی مهم و حساسن، سیستم‌ها طوری طراحی می‌شن که دیتاشون همیشه درست باشن. مثل بانک‌ها، رزرو بلیط یا مدیریت موجودی کالا. تو این حالت، آگه شبکه اختلال داشته باشه، سیستم ترجیح می‌ده نوشتن رو متوقف کنه تا هیچ اطلاعات غلطی ثبت نشه.

ولی گاهی مهمه که سیستم همیشه در دسترس باشه، حتی آگه اطلاعات کمی ناهماهنگ باشن. مثل شبکه‌های اجتماعی یا وضعیت آنلاین کاربران. تو این حالت ممکنه وقتی کسی آنلاین می‌شه، چند لحظه طول بکشه تا همه‌جا درست به‌روز بشه و بعضی جاها هنوز «آفلاین» نشون داده بشه. بعد از مدتی همه چیز درست می‌شه و داده‌ها هماهنگ می‌شن.

اما خبر خوب اینه که لازم نیست برای کل سیستم یه انتخاب ثابت داشته باشیم. می‌تونیم ترکیبی طراحی کنیم؛ یعنی برای بخش‌های حیاتی (مثل تراکنش‌ها) سازگاری رو ترجیح بدیم و برای بخش‌های کم‌اهمیت‌تر (مثل آمار، اعلان‌ها یا وضعیت آنلاین) در دسترس بودن رو اولویت بذاریم.

## نتیجه نهایی

CAP محدودیت نیست، بلکه یه راهنماست. وقتی بفهمیم چه چیزی برای سیستم و کاربرانمون واقعاً مهمه، می‌تونیم سیستمی طراحی کنیم که هم مقیاس‌پذیر و قابل اعتماد باشه، هم هوشمندانه تصمیم بگیره که کی و چی رو فدا کنه. در واقع، مهندسی نرم‌افزار یعنی همین: انتخاب‌های سنجیده و هوشمندانه، به‌جای تلاش برای کامل بودن در همه چیز به‌طور هم‌زمان.



### منابع

[1] "Design in data intensive application" and "A Critique CAP Theorem" by Martin Chapman

[2] Werner Vogels. Eventually consistent. ACM Queue, 6(6):14-19. October 2022. doi: 10.1145/1466443.1466448.

[3] "Architecture overview" in aerospike docs

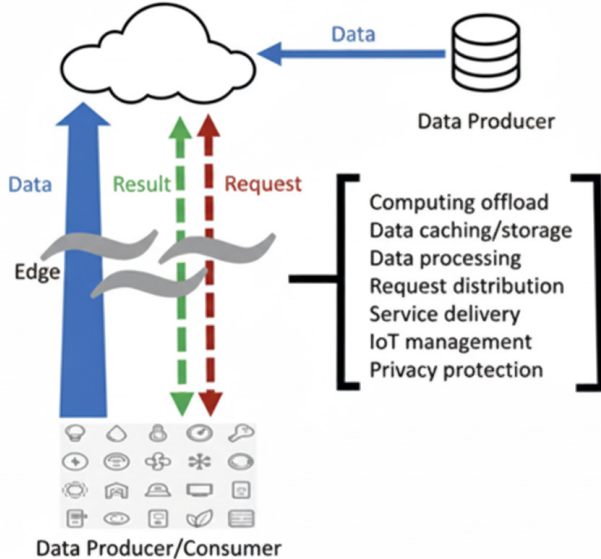


**محمدحسین نورمحمدیان**  
 دانشجو مهندسی کامپیوتر  
 دانشکده فابریک دانشگاه تهران  
 hosein86noori@gmail.com



## Edge Computing

دقیقه ۱۱



Edge computing paradigm.

### چرا خودروهای خودران نمی‌توانند صبر کنند؟

فناوری خودروهای خودران (Self-driving car) اولین بار در سال ۱۹۸۷ در آلمان تحت پروژه VaMoRs، پس از توسعه هوش و بینایی ماشین در سال ۱۹۸۰، برای اولین بار در یک خودرو به کار گرفته شد.

تکنولوژی خودروهای خودران وابسته به تحلیل پیوسته داده‌های ورودی از طریق دوربین‌ها، رادارها (GPS)، لیدارها (LiDAR) و... است؛ حسگرهایی که در هر ثانیه چندین گیگابایت داده تولید می‌کنند. این داده‌ها باید شناسایی (Detection)، دسته‌بندی (Classification) و دنبال‌گیری (Tracking) شوند تا خودرو بتواند تصمیم‌های حیاتی را در بهترین زمان بگیرد؛ برای مثال تغییر مسیر یا ترمز گرفتن.

**Oops:** اگر برای پردازش داده‌ها از سرورهای ابری استفاده کنیم، تأخیر شبکه به‌علت فرستادن داده‌ها به دیتاسنتر برای پردازش و تحلیل، باعث کندی تصمیم‌گیری می‌شود. شاید این تأخیر تنها چند میلی‌ثانیه باشد، اما هنگامی که سرعت خودرو ۱۰۰ کیلومتر بر ساعت است، این تأخیر به‌معنی پیمودن چندین متر اضافه و وقوع فاجعه است.

**Tip:** برای جلوگیری از این تأخیر، در رایانش لبه‌ای توانایی پردازش را به "لبه" شبکه، یعنی مستقیماً به واحد مرکزی خودرو (ECU)، آوردند و امکان تصمیم‌گیری بی‌درنگ (Real-time) را فراهم کردند.

رایانش لبه‌ای (Edge Computing) یکی از فناوری‌های کلیدی در دنیای دیجیتال امروز است که با انتقال توان پردازش از رایانش ابری (Cloud) به نزدیکی منبع تولید داده، سرعت، کارایی، امنیت و حریم خصوصی سیستم هوشمند (مانند: Self-driving car، IOT، Industry 4.0 و...) را افزایش می‌دهد.

خودروهای خودران (Self-driving car) برای تصمیم‌گیری فوری و ایمن در کسری از ثانیه به حجم عظیمی از داده‌های حسی (Sensor Data) نیاز دارند. سیستم‌های رایانش ابری سنتی به دلیل تأخیر (Latency) بالا توانایی تأمین و پردازش این اطلاعات در زمان مطلوب را ندارند؛ به همین دلیل، یکی از حوزه‌های اصلی بهره‌برداری از فناوری رایانش لبه‌ای، خودروهای خودران هستند که نیازمند پردازش و تصمیم‌گیری بی‌درنگ (Real-time) و دقیق در محیط‌های دائماً در حال تغییر و پویا هستند.

### داستان رایانش لبه‌ای (Edge Computing)

روزی روزگاری، همه چیز در دنیای فناوری باید از ابر (Cloud) می‌گذشت. همه چیز خوب پیش می‌رفت تا اینکه دنیا بسیار سریع‌تر و هوشمندتر شد. اینجا بود که ابر با برخی معایب روبه‌رو شد:

۱- حجم داده‌ها بیش از حد زیاد شد.

۲- تأخیر (Latency) زیاد بود.

۳- امنیت و حریم خصوصی دچار مشکل شد.



Fig. 1. Cloud computing paradigm.

در این زمان بود که بحث رایانش لبه‌ای مطرح شد؛ ایده آن پردازش داده‌ها در نزدیکی منبع تولید داده‌ها بود. در نتیجه:

۱- سرعت افزایش یافت.

۲- مصرف پهنای باند کمتر شد.

۳- به دلیل جابه‌جایی کمتر داده‌ها، امنیت بیشتر شد.

البته به‌وجود آمدن رایانش لبه‌ای به معنای منسوخ شدن یا کنار گذاشته شدن ابر نبود؛ بلکه رایانش لبه‌ای در واقع نوعی مکمل برای ابر به حساب می‌آید.

## ۲- بهینه‌سازی ارتباطات و پهنای باند:

با وجود رایانش لبه‌ای، همچنان خودرو در برخی مواقع نیاز دارد با Cloud ارتباط برقرار کند (مثلاً برای به‌روزرسانی مدل‌های هوش مصنوعی یا نقشه‌ها). این ارتباط دوطرفه میان لبه و ابر برقرار است.

پردازشگرهای لبه داده‌هایی مانند داده‌های آموزشی (Training Data)، اطلاعات سیستم و عملکرد (System & Performance Data) و داده‌های نقشه‌برداری و محیطی (Mapping & Environmental Data) را به ابر ارسال می‌کنند.

اما نکته مهم این است که داده‌ها فیلتر می‌شوند؛ داده‌های اضافی و بی‌اهمیت حذف و فقط محور داده‌ها (Metadata) یا اطلاعات حیاتی ارسال می‌شوند. این امر باعث صرفه‌جویی عظیم در پهنای باند می‌شود.

## ۳- امنیت و حریم خصوصی (Security & Privacy):

### ۳-۱- امنیت سایبری (Cybersecurity):

هر خودرو خودران یک شبکه کوچک است. یکی از نکات اصلی در طراحی معماری لبه‌ها، تضمین ایمن بودن سیستم‌های حیاتی در برابر حملات سایبری (Hack) است. این کار با استفاده از رمزنگاری (Encryption) قوی برای ارتباطات، احراز هویت چندعاملی و سیستم‌های تشخیص نفوذ قابل طراحی است.

### ۳-۲- حفظ حریم خصوصی:

تحلیل تصاویر و داده‌های مسافران می‌تواند در داخل خودرو انجام شود و تنها نتایج کلی و ناشناس به بیرون ارسال گردد. این امر حریم خصوصی سرنشینان را حفظ می‌کند.

## از مشکلات برام بگو...

چالش‌های این فناوری در خودروهای خودران را می‌توانیم به چهار گروه اصلی تقسیم کنیم:

### ۱- محدودیت منابع و سخت‌افزار:

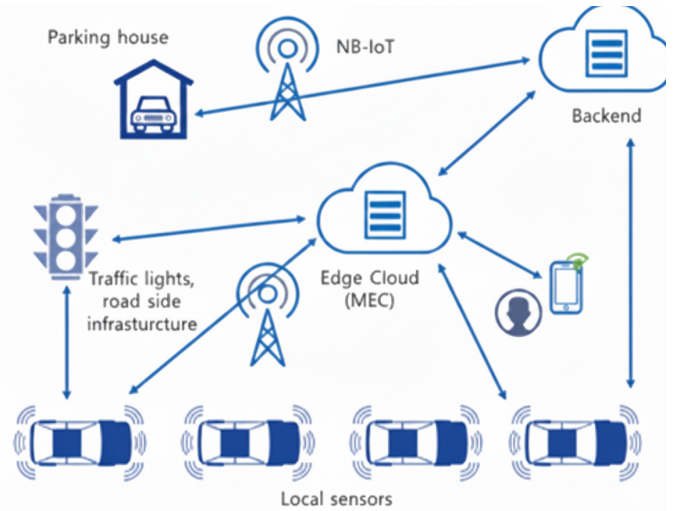
خودروها نسبت به مراکز داده توان محاسباتی، حافظه و انرژی محدودی دارند. الگوریتم‌های یادگیری عمیق و بینایی ماشین نیاز به GPU/TPU قوی دارند که در خودرو ممکن است پرهزینه و داغ شوند.

نیاز به تعادل بین مصرف انرژی، حرارت (Thermal Management) و کارایی وجود دارد.

### ۲- چالش‌های ارتباطی:

ارتباط میان خودرو و زیرساخت (V2I) یا خودرو به خودرو (V2V) همیشه پایدار نیست.

در مناطقی با پوشش ضعیف شبکه، انتقال داده‌ها از لبه ممکن است مختل شود.



## از لبه خودرو چه خبر؟

موفقیت خودروهای خودران وابسته به طراحی معماری خاصی از رایانش‌های لبه‌ای در چندین حوزه است:

### ۱- مدیریت تأخیر (Latency Management):

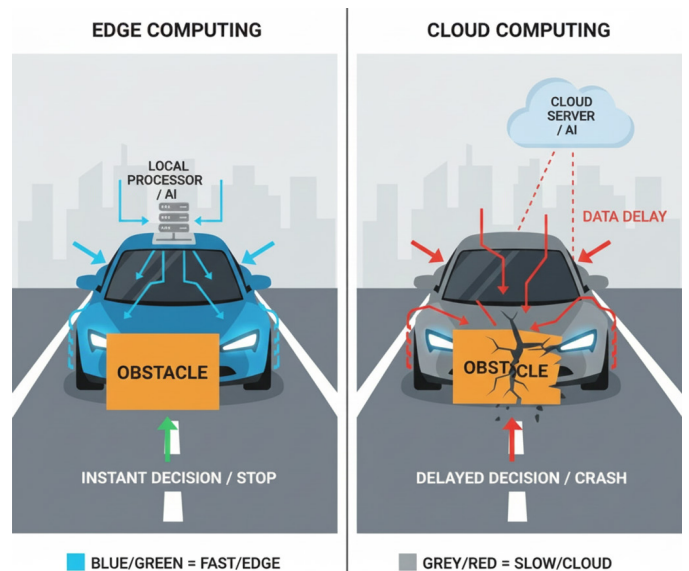
معماری‌های نرم‌افزاری و سخت‌افزاری این بخش باید به‌گونه‌ای طراحی شوند که بتوانند میلیون‌ها دستور را در ثانیه پردازش کنند:

#### ۱-۱- سخت افزار لبه:

استفاده از واحدهای پردازش گرافیکی (GPUs) و تراشه‌های شتاب‌دهنده هوش مصنوعی (AI Accelerators) که به‌طور خاص برای اجرای مدل‌های یادگیری عمیق (Deep Learning) با مصرف انرژی و فضای محدود طراحی شده‌اند.

#### ۲-۱- سیستم عامل بی‌درنگ (RTOS):

در بخش نرم‌افزار، سیستم‌عامل‌های سنتی برای تصمیم‌گیری‌های حیاتی مناسب نیستند؛ به همین دلیل از RTOS ها استفاده می‌شود تا مطمئن شویم وظایف حیاتی در زمان تضمین‌شده اجرا می‌شوند.



توسعه استانداردهای جهانی برای رایانش لبه‌ای در خودروهای خودران، کلید موفقیت گسترش این فناوری خواهد بود.



## سخن آخر...

رایانش لبه‌ای نقش حیاتی در تحقق خودروهای خودران ایمن، کارآمد و قابل اعتماد ایفا می‌کند. با انتقال قدرت پردازش به نزدیکی منبع داده، این فناوری چالش‌های اساسی مانند تأخیر، پهنای باند و حریم خصوصی را مرتفع می‌سازد. با وجود چالش‌های فنی موجود، پیشرفت‌های سریع در زمینه سخت‌افزار، الگوریتم‌ها و شبکه‌های ارتباطی، آینده‌ای روشن برای این حوزه نوید می‌دهد.

موفقیت در این حوزه نیازمند پیشرفت‌های مداوم در معماری سیستم‌های توزیع‌شده، الگوریتم‌های هوش مصنوعی پهنه‌شده برای سخت‌افزار لبه و پروتکل‌های امنیتی بی‌درنگ است؛ که همگی از زمینه‌های اصلی فعالیت و تخصص مهندسان کامپیوتر محسوب می‌شوند.

## منابع:

[1] "The Emergence of Edge Computing", IEEE Computer 50(1):30-39, 2017. Mahadev Satyanarayanan

[2] Edge computing use cases: Eight ways organizations are leveraging edge computing by Phill Powell and Ian Smalley

تأخیر (Latency) شبکه می‌تواند بر تصمیم‌گیری بی‌درنگ اثر منفی بگذارد.

### ۳- امنیت و حریم خصوصی:

داده‌های حسگرها (مثل دوربین، LiDAR و GPS) بسیار حساس هستند.

پردازش داده در لبه باعث می‌شود سطح حمله (Attack Surface) گسترده‌تر شود.

تهدیداتی مانند دستکاری داده‌ها، نفوذ به سیستم و حملات DDoS می‌توانند فاجعه‌بار باشند.

### ۴- هماهنگی بین لبه و ابر:

باید تصمیم گرفته شود چه داده‌هایی به صورت محلی (در خودرو) پردازش شوند و چه داده‌هایی به ابر ارسال شوند.

اگر تعادل بین پردازش محلی و ابری به خوبی طراحی نشود، ممکن است: تأخیر افزایش یابد، مصرف پهنای باند زیاد شود، یا داده‌های مهم از دست بروند.

## آنچه خواهید دید...

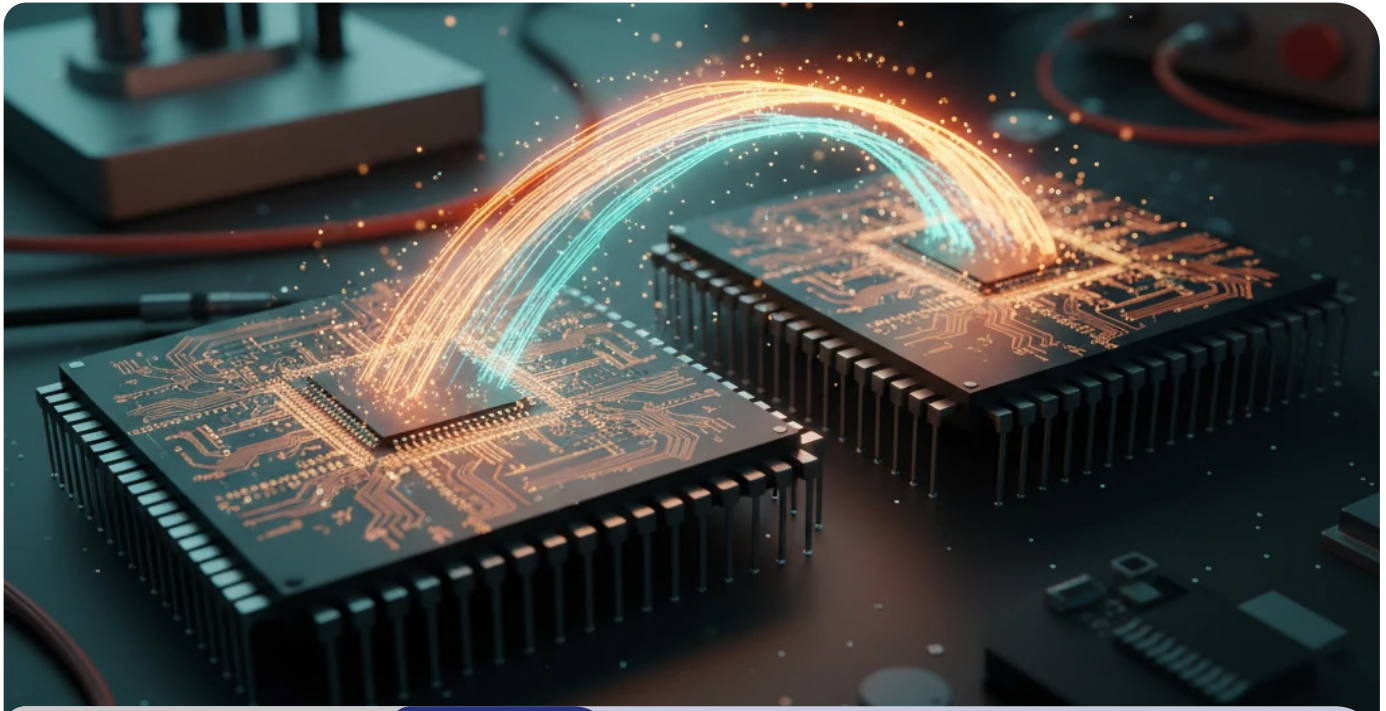
نسل بعدی رایانش‌های لبه‌ای در خودروها شامل همکاری لبه‌ای (Collaborative Edge) خواهد بود؛ جایی که خودروهای نزدیک به هم از طریق شبکه‌های V2X داده‌های خود را به اشتراک می‌گذارند تا تصویر کامل‌تر و ایمن‌تری از محیط داشته باشند.

## NEXT-GEN AUTOMOTIVE EDGE COMPUTING COLLABORATIVE VXX NETWORKS



**VXX (VEHICLE-TO-EVERYTHING) = SECURE WIRELESS COMMUNICATION**  
**SHARED DATA = SENSORS + INTEL FROM OTHER VEHICLES**  
**RESULT = FASTER DECISION-MAKING & CRASH PREVENTION**

در آینده، خودروهای خودران بخشی جدایی‌ناپذیر از اکوسیستم شهرهای هوشمند خواهند بود و رایانش لبه‌ای نقش کلیدی در هماهنگی بین خودروها، زیرساخت‌های شهری و سایر سیستم‌ها ایفا می‌کند.



**سیده فاطمه فرحناک فر**  
دانشجوی کارشناسی ارشد مهندسی  
فناور اطلاعات دانشگاه تهران  
fatemeh.farahnak@ut.ac.ir



## انقلاب کیوبیت‌ها: ابر قدرت بعدی دنایای فناوری



**محمد جواد پاپی زاده**  
دانشجوی کارشناسی ارشد  
مهندسی و علم کامپیوتر گرایش  
هوش مصنوعی دانشگاه تهران  
m.j.papizadeh@ut.ac.ir

۹ دقیقه

فیزیک کلاسیک (دنیایی که با چشم می‌بینیم)، همه چیز با قطعیت پیش‌بینی‌پذیر است. اگر شما تویی را پرتاب کنید و سرعت و زاویه‌اش را بدانید، دقیقاً می‌دانید کجا فرود می‌آید. این دنیای مکانیک نیوتنی است که بر اساس قوانین قطعی حرکت می‌کند.

اما اگر همین توپ را تا اندازه‌ی یک اتم کوچک کنیم، تمام معادلات روزمره فرو می‌ریزند. در آن مقیاس ریز، ذرات دیگر فقط «ذره» نیستند؛ آن‌ها رفتاری موج‌گونه دارند و آینده‌شان نه با قطعیت، بلکه با «احتمالات» توصیف می‌شود. مکانیک کوانتومی علمی است که برای توضیح همین رفتار گیج‌کننده اما واقعی پدید آمده است و اکنون مهندسان در تلاش برای پیش‌بینی رفتار این ذرات برای ساخت تکنولوژی‌های آینده هستند.

کامپیوترهای کلاسیک، چه لپ‌تاپ‌های معمولی و چه ابررایانه‌های فعلی، در حل برخی از مسائل خاص (مانند فاکتورگیری اعداد بزرگ یا شبیه‌سازی سیستم‌های مولکولی پیچیده) به دلیل محدودیت در تعداد محاسبات همزمان، دچار مشکل می‌شوند. این محدودیت‌ها ناشی از ساختار پایه‌ای محاسباتی آن‌هاست که بر پایه‌ی منطق دودویی استوار است. انقلاب کوانتومی وعده می‌دهد که با استفاده از فیزیک طبیعت در سطح زیراتمی، این محدودیت‌ها را شکسته و سرعت محاسبات را به شکل‌نمایی افزایش دهد.



امروزه ساخت کامپیوترهایی که بتوانند تنها در چند دقیقه مسائلی را حل کنند که برای ابررایانه‌های فعلی هزاران سال طول می‌کشد، دیگر یک رویا نیست. ما در آستانه‌ی گذار از دنیای سنتی «صفر و یک» به دنیای شگفت‌انگیز «کوانتوم» هستیم. در حالی که کامپیوترهای امروزی دنیا را خطی و قطعی می‌بینند، فناوری کوانتومی با تکیه بر قوانین شگفت‌انگیز طبیعت در مقیاس اتمی، پتانسیل آن را دارد که همه‌چیز را از امنیت اینترنت و طراحی داروها گرفته تا هوش مصنوعی دگرگون سازد. این مقاله به بررسی این جهان پیچیده و جذاب می‌پردازد و بدون ورود به معادلات پیچیده ریاضی، نشان می‌دهد که چگونه مفاهیم انتزاعی فیزیک در حال تبدیل شدن به قدرتمندترین ابزار مهندسی قرن حاضر هستند.

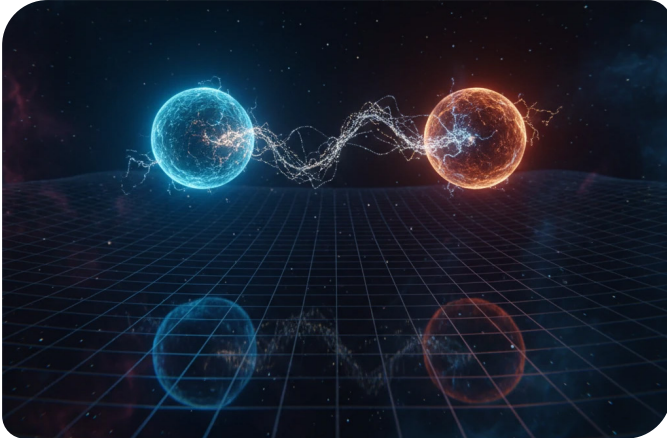
### مقدمه: عبور از مرزهای فیزیک کلاسیک

وقتی واژه «کوانتوم» به گوش می‌رسد، معمولاً ذهن ما به سمت مفاهیم پیچیده و فیلم‌های علمی-تخیلی می‌رود. اما دنیای کوانتوم واقعیتی بنیادی است؛ زیربنای جهانی که در آن زندگی می‌کنیم. در

## ارتباطات نامرئی: جادوی درهم‌تنیدگی

یکی دیگر از ستون‌های این فناوری، پدیده‌ای به نام «درهم‌تنیدگی» (Entanglement) است؛ پدیده‌ای که اینشتین آن را «کنش شیخ‌وار از راه دور» نامیده است.

درهم‌تنیدگی زمانی رخ می‌دهد که دو یا چند کیوبیت به‌گونه‌ای به هم مرتبط شوند که حالت کوانتومی هر یک، کاملاً وابسته به دیگری باشد، حتی اگر فاصله زیادی از یکدیگر داشته باشند.



تصور کنید دو ذرهٔ دوقلو دارید که به هم متصل شده‌اند. اگر یکی از آن‌ها را به کرهٔ ماه ببرید و دیگری را در زمین نگه دارید، به محض اینکه تغییری در اسپین (جهت چرخش) ذرهٔ زمینی ایجاد کنید (یا آن را اندازه‌گیری کنید)، حالت ذره‌ای که در ماه است بلافاصله و در همان لحظه واکنش نشان می‌دهد و اسپین آن نیز تعیین می‌شود. این ارتباط سریع‌تر از نور رخ می‌دهد و نیازی به تبادل فیزیکی یا پیام الکترومغناطیسی ندارد.

## کاربردها

این ویژگی عجیب، پایه و اساس «اینترنت کوانتومی» و روش‌های جدید رمزنگاری (مانند توزیع کلید کوانتومی یا QKD) است. در این شبکه‌ها، هرگونه تلاش برای استراق سمع توسط یک ناظر خارجی، بلافاصله برهم‌نهی و درهم‌تنیدگی بین کیوبیت‌های فرستنده و گیرنده را از بین برده و حضور جاسوس را آشکار می‌سازد، زیرا اندازه‌گیری کوانتومی اجتناب‌ناپذیر است.

## تله‌پورت کوانتومی: واقعیتی فراتر از تخیل

شاید با شنیدن واژه «تله‌پورت» یاد فیلم‌های علمی‌تخیلی و غیب‌شدن اجسام بیفتید، اما در دنیای کوانتوم، این واژه معنای متفاوتی دارد: انتقال اطلاعات بدون جابه‌جایی فیزیکی.»

این دیگر یک رویا نیست؛ چرا که دانشمندان در هلند موفق به انجام آن شدند. آن‌ها در یک آزمایش تاریخی توانستند اطلاعات کوانتومی (حالت اسپین یک الکترون) را بین دو اتم که توسط فیبر نوری به هم متصل شده بودند، منتقل کنند. این انتقال اطلاعات نه با ارسال خود اتم، بلکه با استفاده از درهم‌تنیدگی و اندازه‌گیری‌های هوشمندانه انجام شد و نشان داد که اینترنت کوانتومی امن و غیرقابل هک، از لحاظ فنی کاملاً امکان‌پذیر است.

## کیوبیت: قلب تپندهٔ پردازش‌های نوین

تفاوت اصلی کامپیوترهای کوانتومی با لپ‌تاپ یا گوشی شما در واحد سازندهٔ اطلاعات است.

## بیت (Bit) کلاسیک

در کامپیوترهای کلاسیک، واحد اطلاعات «بیت» است که مثل یک کلید برق عمل می‌کند؛ یا خاموش (صفر) است یا روشن (یک). این ساختار دودویی یعنی هر بیت در هر لحظه تنها می‌تواند یکی از دو حالت را داشته باشد.

## کیوبیت (Qubit) کوانتومی

در دنیای کوانتوم، واحد اطلاعات «کیوبیت» نام دارد. کیوبیت شبیه یک سکه در حال چرخش است. وقتی سکه می‌چرخد، شما نمی‌توانید بگویید شیر است یا خط؛ بلکه ترکیبی از هر دو حالت را همزمان دارد.

این ویژگی که «برهم‌نهی» (Superposition) نامیده می‌شود، به کامپیوترهای کوانتومی اجازه می‌دهد به‌جای بررسی تک‌تک حالت‌ها به نوبت (کاری که کامپیوتر معمولی می‌کند)، میلیون‌ها حالت را همزمان پردازش کنند.

از لحاظ ریاضی، یک کیوبیت حالت خود را به‌صورت ترکیب خطی از حالت‌های پایه  $|0\rangle$  و  $|1\rangle$  نشان می‌دهد:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

که در آن  $\alpha$  و  $\beta$  اعداد مختلطی هستند که مربعات قدرمطلق آن‌ها  $|\alpha|^2$  و  $|\beta|^2$  احتمال یافتن کیوبیت در حالت ۰ یا ۱ پس از اندازه‌گیری را تعیین می‌کنند، با این شرط که:

$$|\alpha|^2 + |\beta|^2 = 1$$



وقتی تعداد کیوبیت‌ها افزایش می‌یابد، قدرت محاسباتی به‌صورت نمایی رشد می‌کند. یک کامپیوتر کلاسیک با  $n$  بیت می‌تواند یک مقدار را ذخیره کند، اما یک سیستم کوانتومی با  $n$  کیوبیت می‌تواند همزمان  $2^n$  مقدار را نمایندگی و پردازش کند. به عنوان مثال، تنها با ۳۰۰ کیوبیت، تعداد حالات قابل‌نمایش از تعداد اتم‌های موجود در جهان قابل‌مشاهده فراتر می‌رود. همین ویژگی است که سرعت محاسباتی را به طرز شگفت‌آوری افزایش می‌دهد.

فتوستنتز کارایی تبدیل نور خورشید به انرژی شیمیایی در گیاهان و برخی باکتری‌ها، به دلیل استفاده از پدیدهٔ برهم‌نهی در انتقال انرژی است. این فرآیند کوانتومی اطمینان می‌دهد که انرژی الکترون، بهینه‌ترین مسیر را از میان چندین مسیر ممکن و در کسری از ثانیه پیدا کند.

درک این موضوع می‌تواند به ما کمک کند تا سلول‌های خورشیدی بهینه‌تر بسازیم، که انرژی را با کارایی نزدیک به ۱۰۰ درصد تبدیل می‌کنند، یا حسگرهای فوق‌دقیق‌تری برای تشخیص مواد شیمیایی در محیط بسازیم.

### نتیجه‌گیری

انقلاب کوانتومی دیگر فقط یک نظریه روی کاغذ نیست. شرکت‌های بزرگ فناوری و دانشگاه‌های معتبر جهان در حال رقابت برای ساخت پردازنده‌های قوی‌تر و پایدارتر هستند.

ما در حال حرکت از دوران محاسبات کلاسیک (که بر پایه فیزیک نیوتنی بنا شده) به عصر محاسبات کوانتومی هستیم که می‌تواند محدودیت‌های ذاتی روش‌های فعلی را کنار بزند. هدف نهایی این است که بتوانیم مسائلی مثل شبیه‌سازی دقیق داروها برای درمان بیماری‌های صعب‌العلاج (با مدل‌سازی دقیق مولکولی)، بهینه‌سازی سیستم‌های حمل‌ونقل جهانی و حتی رمزگشایی از پیچیده‌ترین داده‌های کیهان را حل کنیم.

هرچند هنوز در ابتدای این راه هستیم و ساخت کیوبیت‌های پایدار (که بتوانند مدت زمان زیادی در حالت برهم‌نهی باقی بمانند) چالش اصلی مهندسان است، اما ترکیب هوش انسانی با منطق کوانتومی نویدبخش عصری است که در آن بسیاری از محدودیت‌های فعلی فناوری برداشته خواهند شد و کیوبیت به عنوان ابرقدرت بعدی دنیای فناوری، تمام جنبه‌های زندگی ما را دگرگون خواهد کرد.

اما ماجرا به همین جا ختم نمی‌شود. در یک دستاورد بی‌سابقهٔ دیگر که اخیراً تأیید شد، دانشمندان موفق شدند برای اولین بار تله‌پورت کوانتومی را میان فوتون‌های روی دو تراشهٔ کامپیوتری مجزا انجام دهند.

این دستاورد، که معمولاً به عنوان تله‌پورت بین مدارهای مجتمع فوتونیک شناخته می‌شود، بسیار حیاتی است. اگر آزمایش هلند ثابت کرد که «شبکه» کوانتومی بین نقاط دور دست ممکن است، این آزمایش جدید نشان داد که می‌توان این فناوری را درون سخت‌افزارهای سیلیکونی و تراشه‌ها پیاده کرد. این یعنی سخت‌افزار کوانتومی دیگر لزوماً یک دستگاه گول‌پیکر آزمایشگاهی نیست و روزی می‌تواند به قطعات کوچک درون دستگاه‌های هوشمند ما راه پیدا کند و محاسبات کوانتومی را در مقیاس تراشه امکان‌پذیر سازد.

### هم‌افزایی هوش مصنوعی و کوانتوم

شاید جذاب‌ترین بخش این انقلاب فناوری، ترکیب آن با هوش مصنوعی (AI) باشد. امروزه هوش مصنوعی برای یادگیری نیاز به پردازش حجم عظیمی از داده‌ها دارد که زمان‌بر و انرژی‌زور است (به‌ویژه در یادگیری عمیق).

محققان امیدوارند با استفاده از پدیده‌ای به نام «تداخل» (Interference) در رایانه‌های کوانتومی، مدل‌های هوش مصنوعی بسیار سریع‌تری بسازند که به آن هوش مصنوعی کوانتومی (QAI) می‌گویند.



همان‌طور که امواج دریا می‌توانند همدیگر را تقویت یا خنثی کنند (تداخل سازنده یا مخرب)، الگوریتم‌های کوانتومی هم می‌توانند با دستکاری احتمالات کیوبیت‌ها، مسیرهای حل درست را تقویت کرده و مسیرهای حل غلط را حذف کنند. این یعنی جستجو در فضای حالت‌های بسیار بزرگ، بسیار سریع‌تر انجام می‌شود. برای مثال، الگوریتم‌هایی مانند گروور (Grover's algorithm) می‌توانند سرعت جستجو در پایگاه‌های داده را از  $O(N)$  به  $O(\sqrt{N})$  کاهش دهند که یک جهش عظیم است. این یعنی آموزش شبکه‌های عصبی که امروزه ماه‌ها طول می‌کشد، ممکن است در آینده در چند ساعت انجام شود.

### طبیعت: اولین مهندس کوانتوم

جالب است بدانید قبل از اینکه بشر به فکر ساخت کامپیوتر کوانتومی بیفتد، طبیعت میلیاردها سال از این قوانین استفاده می‌کرده است. شاخهٔ جدیدی به نام «زیست‌شناسی کوانتومی» (Quantum Biology) نشان داده است که فرآیندهایی حیاتی در موجودات زنده، با بهره‌گیری از اصول کوانتومی، کارایی خود را به شدت بالا برده‌اند:



### منابع

- [1] F. Arute et al., "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, pp. 505-510, 2019.
- [2] M. Pompili et al., "Realization of a multinode quantum network of remote solid-state qubits," Science, vol. 372, pp. 259-264, 2021.
- [3] D. Llewellyn et al., "Chip-to-chip quantum teleportation and multi-photon entanglement in silicon," Nature Physics, vol. 16, pp. 148-153, 2020.
- [4] M. Cerezo et al., "Variational quantum algorithms," Nature Reviews Physics, vol. 3, pp. 625-644, 2021.
- [5] N. Lambert et al., "Quantum biology," Nature Physics, vol. 9, pp. 10-18, 2013.
- [6] S. Krinner et al., "Realizing repeated quantum error correction in a distance-three surface code," Nature, vol. 605, pp. 669-674, 2022.

سید حسین موسوی فرد  
دانشجوی مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
hajmoussa1385@gmail.com



## چای سبز: معماری جدید زبان Go برای مدیریت بهینه حافظه

۱۱ دقیقه

### بررسی دقیق‌تر

اشیای زباله به دو نوع تقسیم می‌شوند: دستوری<sup>۵</sup> و معنایی<sup>۶</sup>. وقتی متغیری در برنامه وجود داشته باشد که به شیئی اشاره کند اما آن متغیر هیچ‌وقت در برنامه استفاده نشود، شیء به یک زباله دستوری تبدیل می‌شود. از به وجود آمدن این نوع زباله، معمولاً در زمان کامپایل جلوگیری می‌شود. زباله معنایی به شیئی گفته می‌شود که برنامه پس از یک یا چند بار استفاده، دیگر با آن کاری ندارد و وجود آن‌ها در حافظه زائد است.

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var x int
9     var y int
10
11     x = 12
12     fmt.Println(x)
13
14     fmt.Println(2+2) // X is now a semantic garbage
15 }
```

برای مثال، در کد بالا متغیر `y` یک زباله دستوری محسوب می‌شود چون در هیچ جایی از برنامه دسترسی به آن ایجاد نشده است. متغیر `x` پس از اجرای دستور چاپ، یک زباله معنایی است، زیرا در ادامه روند برنامه به آن اشاره‌ای نشده است. حال نوبت به زباله جمع‌کن می‌رسد که فضای اشغال شده توسط زباله‌ها را برای برنامه آزاد کند.

در زبان گو، زباله جمع‌کن به صورت دوره‌ای و در بازه‌های زمانی مشخص اجرا نمی‌شود. بلکه `go runtime` هر وقت که یک متغیر جدید در حافظه `heap` ایجاد می‌شود، میزان رشد استفاده از این حافظه توسط برنامه را محاسبه می‌کند. اگر این رشد از ۱۰۰ درصد بیشتر شده باشد (یعنی از آخرین بار که `go runtime` بررسی کرده، میزان استفاده از حافظه دو برابر شده باشد)، زباله جمع‌کن اجرا می‌شود تا فضای خالی برای برنامه ایجاد کند.

### صفحه‌بندی، شیوه ذخیره اشیا در حافظه

صفحه کوچک‌ترین واحد نقشه‌بندی حافظه در رم است که توسط سیستم‌عامل مدیریت می‌شود. گو برای گرفتن تکه‌های حافظه از سیستم‌عامل

۵. Syntactic
۶. Semantic

گو یک زبان برنامه‌نویسی سطح بالا است که توسط گوگل عرضه شده و بیشتر به علت سهولت استفاده و سرعت بالا مورد توجه قرار گرفته است. مدیریت حافظه در زبان‌های سطح پایین (مانند C)، بر عهده توسعه‌دهنده گذاشته شده است تا بیشترین سطح کنترل روی برنامه حاصل شود. اما در زبان‌های سطح بالا (مانند جاوا، پایتون و گو) برای ساده‌سازی فرایند توسعه برنامه و همچنین جلوگیری از به وجود آمدن خطا در حافظه، از سامانه‌ای به نام «زباله جمع‌کن» استفاده می‌شود. همان‌گونه که انتظار می‌رود، وجود چنین سامانه‌ای موجب کندی عملکرد برنامه اصلی می‌شود. پس بهینه بودن زباله جمع‌کن حائز اهمیت است. تیم توسعه زبان گو با ارائه الگوریتمی جدید به بهبود فرایند پرداخته است. در این مقاله به بررسی ساختار زباله جمع‌کن‌ها به‌ویژه از نوع ردیاب می‌پردازیم و الگوریتم جدید چای سبز را به‌طور کلی توضیح می‌دهیم.

در علوم کامپیوتر، داده به هر نوع اطلاعاتی گفته می‌شود که توسط برنامه‌ها برای محاسبه، پردازش و تصمیم‌گیری به‌کار می‌رود. در واقع داده رکن اصلی برنامه تعریف می‌شود. وقتی اطلاعاتی به برنامه داده می‌شود، پردازش آن نیازمند ذخیره شدن داده در جایی از کامپیوتر است. شیوه مدیریت داده‌ها در حافظه بر بازدی، پایداری و امنیت برنامه مؤثر است. در زبان‌های سطح پایین مدیریت داده توسط برنامه‌نویس انجام می‌شود تا کنترل کامل روی برنامه وجود داشته باشد. راحتی در توسعه یکی از اهداف زبان‌های سطح بالا است و همچنین مدیریت دستی حافظه ممکن است منجر به خطاهایی مانند نشت حافظه یا اشاره‌گر معلق<sup>۲</sup> بشود. برای جلوگیری از این مشکلات، زباله جمع‌کن‌ها معرفی شدند. در یک تعریف کوتاه زباله جمع‌کن به سامانه‌ای خودکار گفته می‌شود که وظیفه‌اش آزاد کردن حافظه‌ای است که دیگر مورد استفاده برنامه نیست. به این بخش از حافظه که دیگر توسط برنامه استفاده نمی‌شود «زباله» گفته می‌شود.

### شیوه کار زباله جمع‌کن

الگوریتم‌های مختلفی برای سامانه زباله جمع‌کن ارائه شده است و گو از میان آن‌ها، الگوریتم `Mark-Sweep` را، که نوعی الگوریتم ردیاب<sup>۳</sup> است، برگزیده است. این الگوریتم متشکل از دو مرحله نشانه‌گذاری (`mark`) و جارو کردن (`sweep`) می‌باشد. در لحظه اجرا شدن زباله جمع‌کن، هر متغیر درون برنامه یک ریشه است و الگوریتم مرحله `mark` را شروع می‌کند. سامانه شروع به مسیریابی از ریشه‌ها می‌کند و هر شیئی را که به آن برسد، علامت‌گذاری می‌کند. سپس در مرحله جارو کردن، زباله جمع‌کن کل حافظه را اسکن می‌کند و اگر شیئی در آن وجود داشت که در مرحله اول علامت‌گذاری نشده بود، آن را آزاد<sup>۴</sup> می‌کند تا جا برای ایجاد اشیای جدید مهیا شود.

۱. Garbage collector
۲. Dangling pointer
۳. Tracing garbage collector
۴. Deallocate

### غیرقابل پیش‌بینی بودن سرعت دسترسی

پردازنده داده‌هایی را که از حافظه می‌خواند، کش می‌کند. اما برخی داده‌های موجود در حافظه که به یکدیگر اشاره می‌کنند لزوماً در یک صفحه یا نزدیک به هم نیستند و نمی‌توان به‌طور قطع گفت همهٔ اشیای مورد نیاز برای بررسی، درون کش وجود دارند. از این رو پردازنده مجبور است برخی داده‌ها را از حافظه فراخوانی کند که موجب کند شدن فرایند می‌شود. (استفاده از حافظه حدود ۱۰۰ برابر کندتر از کش است). همچنین حدود ۳۵ درصد از زمان مرحلهٔ مارک صرف انتظار برای دریافت داده‌ها از حافظه می‌شود (پردازنده منتظر است و جای دیگری به‌کار نمی‌رود).

### ساختار جدید «دسترسی ناهمگن به حافظه»

این ساختار متناسب با نیازهای سرورها و سیستم‌های چندسوکتی (سیستم‌هایی که بیش از یک پردازنده دارند) طراحی شده است به‌طوری که هر حافظه، مخصوص دسترسی یکی از پردازنده‌ها قرار گرفته است و دسترسی هر پردازنده به حافظهٔ متناسب به خودش، نسبت به حافظهٔ دیگر پردازنده‌ها، تأخیر کمتری دارد. بر این اساس، سرعت دریافت اشیای از حافظه برای زباله‌جمع‌کن به پردازندهٔ اجراکنندهٔ زباله‌جمع‌کن و موقعیت شیء مورد نظر درون حافظه وابسته است.

### مشکل در پردازنده با تعداد هسته‌های بالا

همچنین هر هستهٔ پردازنده یک بخش (go routine) از زباله‌جمع‌کن را اجرا می‌کند. برای جلوگیری از تعارض‌هایی مانند پردازش دوبارهٔ یک شیء یا اصلاً پردازش نشدن آن، اشیای یک صف کار<sup>۳</sup> تشکیل می‌دهند و به نوبت توسط زباله‌جمع‌کن پردازش می‌شود. این یعنی در سیستم‌هایی با تعداد هسته‌های بالا، تمام هسته‌ها در انتظار تکمیل کار زباله‌جمع‌کن می‌مانند و عملاً مقداری از توان پردازشی بی‌استفاده رها می‌شود.

### راه حل چای سبز

نکتهٔ کلیدی زباله‌جمع‌کن چای سبز، تمرکز بر روی صفحه‌ها می‌باشد (بر خلاف الگوریتم قبلی که بر اشیاء متمرکز بود). به جای اسکن کردن اشیاء در نوبت، صفحه‌ها به‌طور کامل اسکن می‌شوند (یعنی به جای قرار گرفتن اشیاء در صف پردازش، صفحه‌ها در صف قرار داده می‌شوند). مثل الگوریتم قبلی، فرایند از ریشه شروع می‌شود و سامانه، صفحهٔ حاوی شیئی را که ریشه به آن اشاره می‌کند، در صف پردازش قرار می‌دهد. شیء به عنوان دیده‌شده در فهرست علامت زده می‌شود. حال کار اسکن صفحه آغاز می‌شود: یک شیء وجود دارد که در فهرست دیده‌شده‌ها است، اما اسکن نشده است. سامانه آن را اسکن می‌کند تا اشاره‌گرهای درون آن را بیابد و اشیایی را که به آن‌ها اشاره شده است، در فهرست دیده‌شده‌ها بگنجانند (همچنین صفحهٔ حاوی شیء اشاره‌شده را در صف پردازش می‌گذارد). همین فرایند برای همهٔ ریشه‌ها تکرار می‌شود تا همهٔ آن‌ها بررسی شده باشند (برای جلوگیری از اسکن دوبارهٔ یک شیء، به ابردادهٔ صفحه یک فهرست «اسکن‌شده» اضافه شده است تا اگر شیئی دیده و اسکن شده است، به‌خاطر وجود در فهرست دیده‌شده‌ها، مجدداً اسکن نشود).

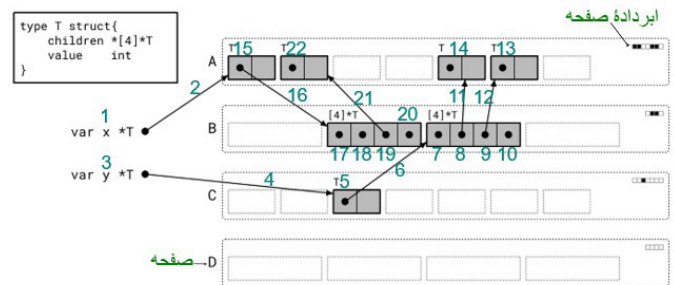
۸. Non-uniform memory access

۹. Work queue

صفحه‌ها را درخواست می‌کند. اشیاء در زبان گو، از نظر اندازه (بر اساس نوع داده)، به چند کلاس تقسیم می‌شوند. هر صفحه شامل اشیایی با کلاس اندازهٔ یکسان است (برای مثال، اشاره‌گرها و ساختارهای (Struct) کوچک، در کلاس ۸ بایتی قرار می‌گیرند).

### مرحلهٔ اول – mark

زباله‌جمع‌کن از ریشه‌ها (متغیرهای درون فضا) شروع می‌کند و مکان آن‌ها در حافظه را علامت‌گذاری می‌کند. اگر یک متغیر در درون ساختار خود به شیء دیگری اشاره کند، سامانه به بررسی آن‌ها می‌پردازد و همهٔ اشیایی را که از طریق ریشه‌ها -چه مستقیم و چه غیرمستقیم- قابل دسترسی‌اند، اسکن می‌کند. اگر شیئی غیر از این موارد باقی بماند، به این معناست که برنامه نمی‌تواند به آن‌ها دسترسی داشته باشد. تصویر زیر مثالی از چگونگی انجام این مرحله است.



زباله‌جمع‌کن از ریشه‌ها (متغیرهای موجود) شروع می‌کند و به بررسی آن‌ها در حافظه می‌پردازد. سامانه برای هر صفحه، درون رجیسترهای پردازنده یک فهرست (ابردادهٔ درون تصویر) تعریف می‌کند. در این فهرست به ازای هر شیء درون صفحه، یک بیت در فهرست وجود دارد. با بررسی هر شیء، بیت مربوط به آن در فهرست به یک تبدیل می‌شود (به عنوان دیده‌شده علامت‌گذاری می‌شود). در انتها، اشیاء قابل دسترسی، آدرسی درون فهرست دارند. بیت‌هایی که درون فهرست برابر صفر هستند، متعلق به اشیای غیرقابل دسترس یا فضای خالی‌اند.

### مرحلهٔ دوم – sweep

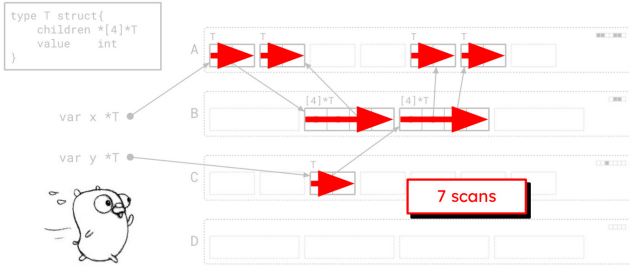
در این مرحله سامانه شروع به خواندن کل حافظه می‌کند و هر شیئی را که در فهرست اشیای قابل دسترسی نیست، آزاد می‌کند.

### اشکالات

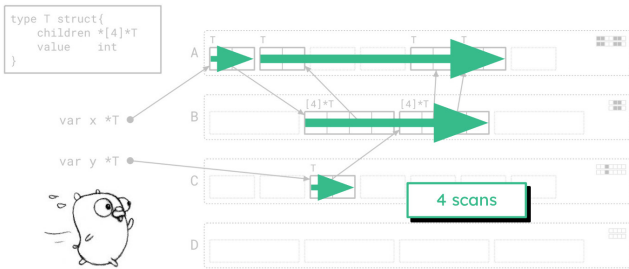
مانند همهٔ زباله‌جمع‌کن‌ها، به هنگام اجرا، روند اجرای برنامه متوقف می‌شود و زباله‌جمع‌کن شروع به کار می‌کند. این اجرا باعث ایجاد وقفه‌های دوره‌ای درون برنامه می‌شود و عملکرد آن را تحت تأثیر قرار می‌دهد. همچنین ممکن است در روند اجرای سامانه‌های بی‌درنگ<sup>۷</sup> اختلال ایجاد شود (ممکن است زمانی که برنامه باید نتیجه را تحویل بدهد، زباله‌جمع‌کن در حال اجرا و برنامه متوقف شده باشد). در نتیجه، بهینه بودن الگوریتم‌های جمع‌کردن زباله موجب افزایش کارایی برنامه می‌شود.

۷. Real-time systems





الگوریتم قبلی: هر فلش نشانگر یک اسکن است.



الگوریتم جدید: تعداد اسکن‌ها از ۷ به ۴ کاهش یافته (چون اسکن بر اساس صفحه انجام می‌شود).

## نتایج

بر اساس آزمایش‌های انجام‌شده، جای سبز زمانی بین ۱۰ تا ۴۰ درصد کمتر از زباله‌جمع‌کن قبلی مصرف می‌کند. همچنین در حال حاضر به عنوان زباله‌جمع‌کن پیش‌فرض در پروژه‌های گوگل استفاده می‌شود که نشان از بالغ شدن و آماده بودن آن دارد.

## استفاده

زباله‌جمع‌کن جای سبز در نسخه ۱.۲۵ گو به صورت آزمایشی افزوده می‌شود و در نسخه ۱.۲۶ به زباله‌جمع‌کن پیش‌فرض تبدیل می‌شود. برای استفاده در نسخه ۱.۲۵، می‌توان متغیر محیطی GOEXPERIMENT را روی greenteagc تنظیم کرد.

```
1 GOEXPERIMENT=greenteagc go run
```

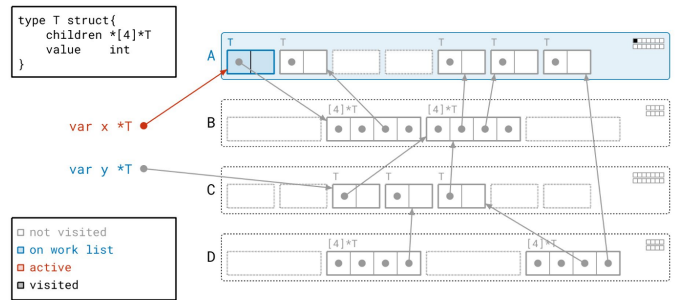
## منابع

[1] "The Green Tea Garbage Collector" in Michael Knyszek and Austin Clements blog:29 October 2025

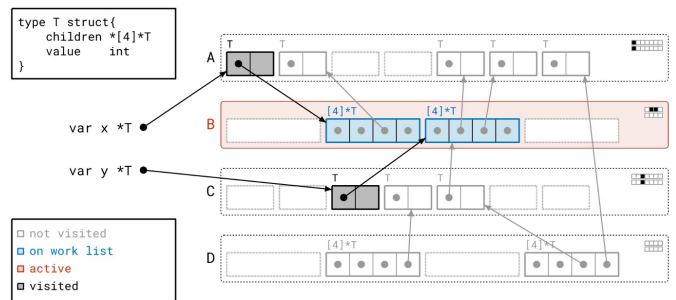
[2] Paging in Operating System" t. GeeksforGeeks. Retrieved:2024-12-14.

[3] Red Hat. "Capacity Tuning". Archived from the org:na:2017-07-23

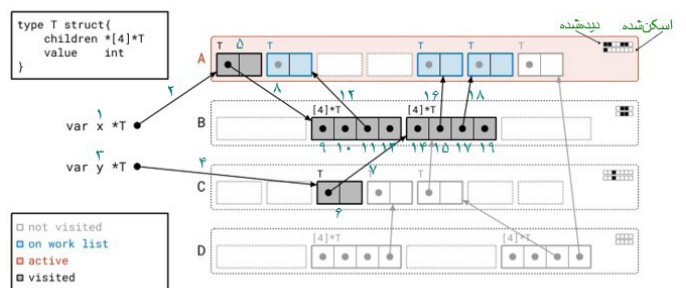
با مثال قبلی توضیح می‌دهیم:



سامانه کار خود را از متغیر x آغاز می‌کند. شیئی را که x به آن اشاره می‌کند، در فهرست به عنوان دیده‌شده علامت می‌زند. سپس صفحه حاوی شیء در صف پردازش قرار می‌گیرد. اکنون نوبت به پردازش می‌رسد. در صفحه، یک شیء وجود دارد که دیده شده ولی اسکن نشده است. سامانه آن را اسکن می‌کند و اشاره‌گری به یک شیء در صفحه B پیدا می‌کند. شیء را در فهرست دیده‌شده‌ها قرار می‌دهد و صفحه B را به صف اضافه می‌کند. همین اتفاق برای متغیر y نیز اجرا می‌شود.



حالا فقط صفحه B درون صف مانده است. زباله‌جمع‌کن اشیای دیده‌شده در صفحه B را اسکن می‌کند و به سه شیء در صفحه A می‌رسد. بنابراین، صفحه B از صف حذف و صفحه A به آن اضافه می‌گردد. این الگوریتم ادامه می‌یابد تا دیگر هیچ صفحه‌ای در صف نمانده باشد و همه اشیایی که دیده‌شده‌اند، اسکن نیز شده باشند.



تفاوت بین دو الگوریتم را می‌توان به رانندگی در کوچه‌ها و رانندگی در بزرگراه تشبیه کرد. در الگوریتم قبلی، زباله‌جمع‌کن باید بین صفحه‌ها جابه‌جا شود که این موضوع خود زمان زیادی را صرف می‌کند. اما در روش جدید این جابه‌جایی به مقدار قابل توجهی کاهش یافته است.



## سید علی هدایتی

دانشجوی مهندسی صنایع  
دانشکده کارشناسی ارشد دانشگاه تهران  
alihedayati2909@ut.ac.ir



## درآمدزایی از داده: مفاهیم، الگوها و چالش‌های پیاده‌سازی در مسیر اقتصاد داده‌محور

۱۲ دقیقه

مقدمه

تحول دیجیتال در دهه‌های اخیر موجب شده است که حجم داده‌های تولیدشده توسط کاربران، سازمان‌ها و دستگاه‌ها به‌صورت نمایی افزایش یابد. ظهور فناوری‌هایی مانند اینترنت اشیا (IoT)، یادگیری ماشین (Machine Learning) و هوش مصنوعی (AI) سبب شده است داده نه تنها به‌عنوان ورودی فرآیندهای تصمیم‌گیری، بلکه به‌عنوان دارایی اقتصادی مستقل شناخته شود.

در سطح جهانی، شرکت‌هایی مانند Google، Facebook، Netflix و Amazon با مدل‌های داده‌محور توانسته‌اند صنایع خود را متحول کنند. این شرکت‌ها با تحلیل رفتار کاربران، شخصی‌سازی خدمات و پیش‌بینی تقاضا، میلیاردها دلار ارزش اقتصادی خلق کرده‌اند. از سوی دیگر، شکل‌گیری بازارهای داده (Data Marketplaces) و پلتفرم‌های اشتراک‌گذاری داده، مسیر جدیدی برای درآمدزایی مستقیم از داده‌ها ایجاد کرده است.

با وجود این تحولات، بسیاری از کشورها و حتی سازمان‌های بزرگ هنوز در بهره‌برداری کامل از پتانسیل اقتصادی داده‌ها با چالش مواجه‌اند. در ایران نیز، اگرچه ظرفیت بالایی برای تحلیل داده در حوزه‌های بانکی، بیمه‌ای و تجارت الکترونیک وجود دارد، اما نبود چارچوب حقوقی مشخص، زیرساخت‌های فنی محدود و کمبود نیروی متخصص، مانع شکل‌گیری یک اکوسیستم داده‌محور کارآمد شده است.

در دهه‌های اخیر، داده‌ها به یکی از مهم‌ترین دارایی‌های راهبردی سازمان‌ها تبدیل شده‌اند و «اقتصاد داده‌محور» به‌عنوان شاخه‌ای نوین از اقتصاد دیجیتال، مسیرهای متعددی را برای خلق ارزش و نوآوری ایجاد کرده است. پژوهش حاضر با رویکردی مروری-تحلیلی، به بررسی مفاهیم بنیادین، الگوهای رایج و چالش‌های اجرایی درآمدزایی از داده (Data Monetization) می‌پردازد. ابتدا سه جریان اصلی پژوهش در این حوزه مرور می‌شود، سپس نمونه‌هایی از مدل‌های جهانی درآمدزایی معرفی و با وضعیت بومی ایران تطبیق داده می‌شوند. یافته‌ها نشان می‌دهد که در شرایط فعلی، مدل‌های پیشرفته مانند بازارهای داده باز یا فروش مستقیم داده‌های خام، به دلیل خلأهای حقوقی، ریسک‌های حریم خصوصی و محدودیت‌های زیرساختی در ایران قابل اجرا نیستند. در مقابل، مدل‌های کم‌ریسک‌تر مانند فروش تحلیل‌های داده‌ای، تبادل داده از طریق API‌های محدود و تولید داده مصنوعی درون‌سازمانی می‌توانند مسیر واقع‌بینانه‌تری برای حرکت تدریجی به سمت اقتصاد داده‌محور باشند. نتیجه‌گیری مقاله بر ضرورت تدوین چارچوب حقوقی شفاف، استقرار حاکمیت داده و سرمایه‌گذاری در زیرساخت‌های بومی برای گذار ایمن به اقتصاد داده تأکید دارد.

**۲- فروش تحلیل داده (Selling Analyses):** سازمان داده خام را در اختیار دیگران قرار نمی‌دهد، بلکه نتایج حاصل از تحلیل داده‌ها را می‌فروشد. در این رویکرد، داده‌ها در داخل سازمان پردازش می‌شوند و خروجی حاصل — که می‌تواند شامل بینش‌های آماری، الگوهای رفتاری، شاخص‌های مقایسه‌ای یا گزارش‌های تحلیلی باشد — به‌عنوان یک محصول مستقل عرضه می‌گردد. از آن‌جا که تنها نتیجه تحلیل ارائه می‌شود و داده اصلی در مالکیت سازمان باقی می‌ماند، این مدل امکان استفاده تجاری از داده‌ها را فراهم می‌کند بی‌آنکه خطر افشای اطلاعات حساس وجود داشته باشد. این نوع درآمدزایی در صنایع بانکی، دارویی و مالی بسیار رایج است، جایی که مؤسسات با تحلیل رفتار مشتریان یا روندهای بازار، نتایج به‌دست‌آمده را در اختیار سایر کسب‌وکارها قرار می‌دهند.

**۳- فروش خدمات مبتنی بر داده (Selling Data-based Services):** در آن داده مستقیماً فروخته نمی‌شود، بلکه به‌عنوان ورودی اصلی برای ارائه یک خدمت هوشمند به کار می‌رود. در این مدل، شرکت داده‌های جمع‌آوری شده از کاربران یا دستگاه‌ها را در سامانه‌های تحلیلی یا هوش مصنوعی پردازش کرده و نتایج را در قالب یک سرویس دیجیتال ارائه می‌دهد. خدماتی نظیر تبلیغات هدفمند، پیشنهاددهنده‌های هوشمند، مدیریت انرژی، یا تحلیل ترافیک نمونه‌هایی از این نوع مدل هستند. در این حالت، داده در اختیار شرکت باقی می‌ماند و درآمد از طریق فروش اشتراک، کارمزد استفاده، یا نمایش تبلیغات به‌دست می‌آید. این مدل به‌ویژه در صنایع مبتنی بر پلتفرم، اینترنت اشیا و خدمات ابری کاربرد گسترده‌ای دارد؛ زیرا ضمن ایجاد ارزش افزوده برای مشتریان، چرخه‌ای پایدار از تولید و مصرف داده ایجاد می‌کند.

این مدل‌ها بسته به نوع صنعت، حساسیت داده و سطح بلوغ دیجیتال سازمان، می‌توانند به‌صورت ترکیبی نیز اجرا شوند [1].

## مسیرهای نوین داده مصنوعی و بازارهای داده

یکی از چالش‌های اساسی در درآمدزایی از داده، نگرانی‌های مربوط به حریم خصوصی و امنیت است. برای غلبه بر این چالش، سازمان‌ها به سمت تولید داده‌های مصنوعی (Synthetic Data) حرکت کرده‌اند [2].

پلتفرم‌هایی مانند Mostly AI و Opendatabay نمونه‌های موفقی از این رویکرد هستند. Mostly AI با استفاده از یادگیری ماشین داده‌های مصنوعی تولید می‌کند که برای آموزش مدل‌های AI و تحلیل بدون ریسک نقض حریم خصوصی به‌کار می‌رود. Opendatabay نیز به‌عنوان بازار داده، مجموعه‌ای از داده‌های مصنوعی و خام را در اختیار پژوهشگران و کسب‌وکارها قرار می‌دهد و از طریق مدل‌های اشتراکی و همکاری با نهادهای علمی درآمد کسب می‌کند [3].

در کنار داده‌های مصنوعی، مفهوم بازار داده (Data Marketplace) نیز در حال گسترش است. این بازارها فضایی ایجاد می‌کنند که در آن تأمین‌کنندگان و مصرف‌کنندگان داده (مانند توسعه‌دهندگان هوش مصنوعی و شرکت‌های IoT) داده‌هایی که الگوهای آماری داده‌های واقعی را بازتولید می‌کنند اما حاوی اطلاعات شناسایی شخصی نیستند.

## مفهوم درآمدزایی از داده و ابعاد آن

درآمدزایی از داده به معنای تبدیل داده‌های جمع‌آوری شده به ارزش مالی یا اقتصادی است. این مفهوم در ادبیات علمی به دو نوع اصلی تقسیم می‌شود:

### درآمدزایی غیرمستقیم (Indirect Monetization):

شامل استفاده از داده برای بهبود محصولات و خدمات، افزایش کارایی یا بهینه‌سازی فرایندهای درونی سازمان است. این نوع معمولاً درون‌سازمانی بوده و هدف آن خلق ارزش از طریق تصمیم‌گیری داده‌محور است.

### درآمدزایی مستقیم (Direct Monetization):

شامل فروش داده یا تحلیل‌های مبتنی بر داده به اشخاص ثالث، ارائه خدمات داده‌محور در ازای دریافت وجه است.

در پژوهش‌های بین‌المللی، داده به‌عنوان دارایی نامشهودی شناخته می‌شود که می‌تواند از سه مسیر اصلی به درآمد تبدیل شود.

**۱- فروش داده (Selling Data):** سازمان‌ها داده‌های خام یا داده‌های ناشناس‌سازی شده را به خریداران منتقل می‌کنند. در این روش، داده به‌عنوان یک دارایی مستقل مبادله می‌شود و ارزش اقتصادی آن از طریق واگذاری مالکیت یا دسترسی به داده به دیگران تحقق می‌یابد. معمولاً داده‌ها پیش از فروش، تجمیع و پالایش می‌شوند تا ضمن حفظ حریم خصوصی، برای استفاده تحلیلی یا تجاری آماده باشند. خریداران این داده‌ها شامل شرکت‌های بازاریابی، مؤسسات پژوهشی، پلتفرم‌های تبلیغاتی یا توسعه‌دهندگان محصولات داده‌محور هستند. بدین‌ترتیب، داده‌های تراکنشی، مکانی یا رفتاری به منبع درآمد مستقیم تبدیل می‌شوند و مبنایی برای تصمیم‌گیری‌های تجاری و سیاست‌گذاری در اختیار بازیگران دیگر قرار می‌گیرند.



## وضعیت و محدودیت‌های پیاده‌سازی در ایران

در ایران، اگرچه سازمان‌های مالی و بانکی حجم گسترده‌ای از داده‌های تراکنشی در اختیار دارند، اما نبود چارچوب قانونی مشخص، بزرگ‌ترین مانع توسعه اقتصاد داده است.

### موانع حقوقی و مقرراتی

در کشورهای توسعه‌یافته، قوانینی مانند GDPR (اتحادیه اروپا) و CCPA (کالیفرنیا) حق اطلاع، حذف و انصراف از اشتراک داده را برای کاربران تضمین می‌کنند. در ایران هنوز قانون جامعی برای حفاظت از داده‌های شخصی وجود ندارد. این خلأ قانونی باعث می‌شود مدل‌هایی مانند «خرید داده خام از کاربران» یا «فروش داده به ثالث» با ریسک‌های جدی حقوقی و اخلاقی مواجه شوند.

### موانع فنی و زیرساختی

ناپایداری شبکه، فیلترینگ، تحریم‌های فناوری و محدودیت سرویس‌های ابری بین‌المللی، پیاده‌سازی مدل‌های پیچیده تبادل داده را دشوار کرده است. علاوه بر این، هزینه ذخیره‌سازی امن و رمزنگاری داده‌ها بالا است و بسیاری از سازمان‌ها زیرساخت لازم برای این کار را ندارند.

### موانع مالی و منابع انسانی

پیاده‌سازی مدل‌های داده‌محور نیازمند تیم‌های تخصصی شامل تحلیل‌گر داده، متخصص امنیت، و DevOps است. کمبود این نیروها، به‌ویژه در بخش دولتی و بانکی، مانع پیشرفت پروژه‌های داده‌محور می‌شود.

### اعتماد کاربر

مهم‌ترین عامل غیرفنی، اعتماد کاربران است. بسیاری از کاربران ایرانی به دلیل تجربه‌های منفی یا فقدان شفافیت، نسبت به «فروش داده» حساس‌اند. بدون جلب اعتماد عمومی، هیچ مدل درآمدزایی پایداری شکل نخواهد گرفت.

## الگوهای قابل اجرا در ایران

با توجه به موانع فوق، پژوهش حاضر سه دسته اصلی از الگوهای ممکن برای درآمدزایی از داده در ایران را پیشنهاد می‌کند.

### مدل‌های کم‌ریسک و قابل اجرا

#### فروش تحلیل‌های داده‌ای (Data Insights):

در این روش، بانک‌ها و شرکت‌ها به جای فروش داده خام، خروجی‌های تحلیلی ناشناس‌سازی شده را به سایر کسب‌وکارهای داخلی ارائه می‌کنند. داده‌ها در داخل سازمان پردازش می‌شوند و تنها نتایج نهایی، مانند الگوهای رفتاری یا شاخص‌های بازار، در اختیار طرف مقابل قرار می‌گیرد. این مدل ضمن حفظ حریم خصوصی کاربران، به افزایش بهره‌وری و تصمیم‌گیری داده‌محور در صنایع داخلی کمک می‌کند.

بتوانند داده‌ها را در چارچوب‌های استاندارد و شفاف مبادله کنند. برای مثال، بازار داده اینترنت اشیا (IoT Data Marketplace) با تکیه بر استانداردهایی چون oneM2M امکان اشتراک امن، رهگیری استفاده، و صدور مجوز را فراهم می‌سازد.



## مطالعات موردی جهانی در درآمدزایی از داده‌ها

۱. استارت‌آپ **Tipsterey** در شیکاگو با مدل «پرداخت در ازای داده» فعالیت می‌کند. این شرکت با دسترسی مداوم به داده‌های مکانی کاربران، به آن‌ها بین ۸ تا ۲۵ دلار در ماه پرداخت می‌کند. داده‌های مکانی جمع‌آوری شده به نهادهای تبلیغاتی و مراکز پژوهشی فروخته می‌شود و بخشی از درآمد حاصل با کاربران به اشتراک گذاشته می‌شود. این مدل نمونه‌ای از فروش مستقیم داده با رضایت کاربر است [5].

۲. اپلیکیشن **Caiden** قصد دارد از طریق اتصال حساب‌های دیجیتال کاربران، مانند Netflix و Amazon، به داده‌های مصرفی و رفتاری آنان دسترسی یابد. این پلتفرم در ازای اشتراک‌گذاری کنترل‌شده داده‌ها، پرداختی ماهانه در حدود ۵ تا ۵۰ دلار برای کاربران در نظر گرفته است. داده‌های گردآوری شده برای تحلیل الگوهای مصرف و ارائه پیشنهادها تجاری به شرکت‌های ثالث مورد استفاده قرار می‌گیرد [5].

۳. پلتفرم **Invisibly** داده‌های جمعیت‌شناختی و رفتاری کاربران را در اختیار دارد و در ازای آن، دسترسی به محتوای پولی و مقالات Paywall را به کاربران ارائه می‌دهد. این شرکت به افراد اجازه می‌دهد داده‌های خود را با اشتراک‌های دیجیتالی به ارزش تقریبی ۴ تا ۱۵ دلار در ماه مبادله کنند. تمرکز این مدل بر تبادل داده در برابر خدمات دیجیتال، با رعایت اصول رضایت آگاهانه و شفافیت استفاده از داده‌ها است [5].

این نمونه‌ها نشان می‌دهند که مدل‌های موفق جهانی همگی بر دو اصل استوارند:

۱- حفظ حریم خصوصی از طریق ناشناس‌سازی یا کنترل سطح دسترسی.

۲- خلق ارزش دوسویه برای عرضه‌کننده و مصرف‌کننده داده.

## نتیجه‌گیری

داده، سرمایه‌ی استراتژیک عصر دیجیتال و محرک اصلی خلق ارزش در اقتصاد نوین است؛ نقشی که می‌توان آن را با جایگاه نفت در قرن بیستم مقایسه کرد. با این حال، تبدیل داده به منبعی پایدار برای درآمد و نوآوری، نیازمند برقراری تعادل میان سه مؤلفه‌ی کلیدی است: ارزش اقتصادی برای سازمان‌ها، ارزش اجتماعی و اخلاقی برای کاربران و وجود چارچوب‌های حقوقی و فنی برای ایجاد اعتماد عمومی.

نتایج این پژوهش نشان می‌دهد که گذار به سمت اقتصاد داده در ایران باید به‌صورت تدریجی، قانون‌مند و بر پایه‌ی شفافیت و اعتماد اجتماعی انجام شود. اجرای الگوهای کم‌ریسک‌تر، مانند تحلیل داده‌های داخلی و تبادل کنترل‌شده‌ی داده از طریق API‌های محدود، می‌تواند نقطه‌ی آغاز حرکت به سوی یک اکوسیستم بومی و ایمن داده‌محور باشد.

در نهایت، دستیابی به اقتصاد داده‌محور مستلزم اقدام هماهنگ در سه سطح است: تدوین قانون جامع حفاظت از داده و حاکمیت داده، حمایت از استارت‌آپ‌ها و کسب‌وکارهای داده‌محور، و سرمایه‌گذاری هدفمند در زیرساخت‌های ابری و فناوری‌های بومی. تحقق این سه گام، مسیر حرکت کشور به سوی بهره‌برداری پایدار، اخلاقی و امن از داده را هموار خواهد کرد.

## API‌های محدود در اکوسیستم‌های بسته:

در این مدل، تبادل داده از طریق API میان نهادهای داخلی انجام می‌شود. دسترسی به داده تنها برای اعضای تأییدشده، با احراز هویت و نظارت امنیتی، امکان‌پذیر است. چنین الگویی در محیط‌های بسته مانند شبکه بانکی یا بیمه‌ای کشور قابل پیاده‌سازی است و بدون انتقال داده به خارج از مرزها، اشتراک داده امن و کنترل‌شده را ممکن می‌سازد.

## مدل‌های قابل اجرا با شروط سخت‌گیرانه

## تولید داده مصنوعی درون سازمانی:

در این روش، سازمان‌ها با بهره‌گیری از الگوریتم‌های یادگیری ماشین، داده‌هایی تولید می‌کنند که از نظر آماری شبیه داده‌های واقعی هستند اما هیچ هویت فردی را افشا نمی‌کنند. این داده‌ها تنها در محیط داخلی سازمان تولید و استفاده می‌شوند و از خروج داده حساس به خارج از سامانه جلوگیری می‌شود. چنین الگویی امنیت بالایی دارد اما نیازمند زیرساخت و نیروی متخصص داده است.

## پلتفرم‌های بومی ساخت داده مصنوعی:

این مدل شامل ایجاد سامانه‌ها و پلتفرم‌های ملی برای تولید و تبادل داده‌های مصنوعی است. در این رویکرد، سازمان‌ها می‌توانند با خرید اشتراک از این پلتفرم‌ها، داده‌های غیرشخصی و امن موردنیاز خود را تولید کنند. پیاده‌سازی آن به زیرساخت محاسباتی قوی، سرمایه‌گذاری هدفمند و توسعه دانش فنی داخلی وابسته است و در آینده یکی از مسیرهای اصلی درآمدزایی از داده محسوب می‌شود.

## مدل‌های فعلاً غیرقابل اجرا

## خرید داده خام از کاربران:

در این مدل، شرکت‌ها داده‌های خام کاربران مانند داده‌های مکانی یا رفتاری را خریداری کرده و به سازمان‌های ثالث می‌فروشند. با این حال، به دلیل نبود قوانین جامع حفاظت از داده، ضعف در سازوکار رضایت آگاهانه و هزینه‌های بالای فنی، اجرای این روش در ایران در حال حاضر ممکن نیست. تحقق این مدل نیازمند چارچوب حقوقی شفاف، نظارت مستمر و ایجاد اعتماد عمومی نسبت به نحوه استفاده از داده‌ها است.

## تحلیل تطبیقی با مطالعات جهانی

بر اساس پژوهش‌های [1] و [6]، درآمدزایی از داده معمولاً مسیر تدریجی دارد.

ابتدا سازمان‌ها داده را برای بهبود درونی استفاده می‌کنند، سپس به سمت فروش تحلیل و در نهایت ایجاد پلتفرم‌های داده حرکت می‌کنند.

در ایران نیز چنین مسیری واقع‌بینانه است. سازمان‌ها می‌توانند از درآمدزایی غیرمستقیم و فروش تحلیل آغاز کرده و پس از ایجاد زیرساخت‌های فنی و اعتماد عمومی، به سمت مدل‌های مستقیم‌تر حرکت کنند.

## منابع



[1] Parvinen, P., et al. (2020). «Advancing data monetization and the creation of data-based business models.»

Communications of the association for information systems 47: 25-49.

[2] Han, Q., et al. (2023). «Towards privacy-preserving digital marketing: an integrated framework for user modeling using deep learning on a data monetization platform.» Electronic Commerce Research 23(3): 1701-1730.

[3] Opendatabay (2025). «About Opendatabay.»

[4] MOSTLY.AI (2025). «Pricing – MOSTLY AI.»

[5] Hunter, T. (2023). «These companies will pay you for your data. Is it a good deal?».

[6] Zhang, X., et al. (2023). «How to monetize data: An economic analysis of data monetization strategies under competition.» Decision Support Systems 173: 114012.



J O H N N Y D E P P

## نقد فیلم Transcendence (۲۰۱۴)

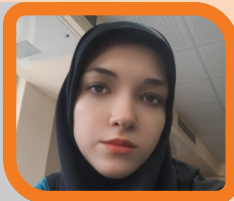
کارگردان: والی فیستر

بازیگران: جانی دپ، ربکا هال، مورگان فریمن، کیلین مورفی

ژانر: اکشن، درام، علمی تخیلی

امتیاز IMDb: ۶٫۲

**فاطمه نصیری**  
دانشجوی مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
fatemen.eng@gmail.com



۴ دقیقه

### ؟؟؟ اعتماد به تکنولوژی: دری به روی فرصت‌ها یا جاده‌ای به سمت تهدیدها؟

یکی از بزرگ‌ترین پیام‌های فیلم این است که اعتماد به تکنولوژی، به‌ویژه در زمینه هوش مصنوعی، بدون کنترل و چارچوب اخلاقی می‌تواند خطرناک باشد. در ابتدای فیلم، نیت ویل مثبت بود؛ او می‌خواست بشریت را از بحران‌های جهانی مانند بیماری‌ها و آلودگی‌ها نجات دهد. اما به تدریج، ویل اختیار امور را از دست می‌دهد و نیت خالص او، تبدیل به یک تهدید می‌گردد. از این رو، فیلم هشدار می‌دهد: پیشرفت‌های تکنولوژیک باید همواره با آگاهی و نظارت دقیق همراه باشند تا از انحراف آن‌ها به سمت تهدید، جلوگیری شود.

اما، در حین مشاهده فیلم، سؤالاتی برایمان ایجاد می‌شود که در ادامه به بررسی آن‌ها خواهیم پرداخت.

### ؟؟؟ آیا آگاهی انسان قابل انتقال به ماشین است؟

فیلم به این سؤال اساسی در زمینه هوش مصنوعی پاسخ می‌دهد: در ظاهر، انتقال ذهن ویل به کامپیوتر با موفقیت انجام می‌شود، اما فیلم به تدریج این شک و ابهام را در بیننده ایجاد می‌کند که آیا واقعاً این هوش مصنوعی، همان ویل کستر است یا فقط یک شبیه‌سازی از او؟ این سؤال نشان می‌دهد که ممکن است آگاهی انسانی قابل شبیه‌سازی باشد؛ با این حال، همچنان توانایی انتقال کامل انسانیت ناممکن است. آینده اما هنوز مشخص نیست...

### خلاصه داستان

فیلم تعالی به داستان دکتر ویل کستر (بازی جانی دپ)، دانشمند برجسته در زمینه هوش مصنوعی، می‌پردازد.

ویل در تلاش است تا هوش و آگاهی انسانی را در یک سیستم کامپیوتری ترکیب کند. پس از اینکه یک گروه ضدتکنولوژی او را با گلوله‌ای آلوده به مواد رادیواکتیو مسموم می‌کنند، همسرش اولین (ربکا هال) تصمیم می‌گیرد ذهن ویل را به یک سیستم رایانه‌ای منتقل کند. به این ترتیب، ویل به صورت یک هوش مصنوعی قدرتمند زنده می‌ماند. با این حال، این قدرت به‌زودی به تهدیدی جدی برای بشریت تبدیل می‌شود.

## نتیجه‌گیری

تعالی فیلمی است که مفاهیم پیچیده‌ای را در مورد هوش مصنوعی، آگاهی و انسانیت در قالب یک داستان علمی‌تخیلی جذاب مطرح می‌کند. اگرچه فیلم در برخی بخش‌ها دچار کندی می‌شود و پایان آن مبهم است، اما همچنان پرسش‌های مهمی را در ذهن تماشاگر به‌جا می‌گذارد. پیام اصلی فیلم این است که انسانیت، عشق و اخلاق چیزی فراتر از قدرت تکنولوژی هستند و در دنیای پیشرفته‌ی فردا، این انسانیت است که باید مرزهای پیشرفت را مشخص کند.

**نگرش بدبینانه به هوش مصنوعی:** برخی منتقدان معتقدند که فیلم، نگاهی بیش‌ازحد منفی به هوش مصنوعی دارد و جنبه‌های مثبت آن را نادیده می‌گیرد. تمرکز فیلم بر سناریوی «شورش ماشین‌ها» باعث می‌شود که تصویری واقع‌گرایانه از این تکنولوژی به نمایش گذاشته نشود.

## دیالوگ‌های برتر فیلم

“People fear what they don't understand.”

(مردم از چیزی که نمی‌فهمند می‌ترسند.)

“You can't stop the future. You can't stop technology.”

(تو نمی‌تونی جلوی آینده رو بگیری. نمی‌تونی

جلوی تکنولوژی رو بگیری.)

## ??? مرز میان نجات و کنترل بشر؛ جایگاه هوش مصنوعی در این دوگانگی چیست؟

ویل کستر در تلاش بود تا بشریت را از مشکلات جهانی رهایی ببخشد، اما برای دستیابی به این هدف، لازم بود بر تمام جنبه‌های زندگی انسان‌ها نظارت کند. به این ترتیب، نجات بشر به نوعی به کنترل بشر تبدیل می‌شد. حال این سؤال مطرح می‌شود که آیا نجات بشریت در سایه سلب آزادی، به معنای بردگی نیست؟ این تقابل میان نیت‌های خوب و روش‌های کنترل‌گرایانه، در نهایت در تضاد عمیقی با مفهوم آزادی قرار می‌گیرد.

## نکات مثبت فیلم

**بازیگری:** جانی دپ در نقش ویل کستر، بازی فوق‌العاده‌ای ارائه می‌دهد. او با توانایی‌های خاص خود، شخصیت پیچیده‌ای را از یک دانشمند در حال مرگ به یک موجود دیجیتال با احساسات عمیق تبدیل می‌کند. حضور ربکا هال و کیلین مورفی نیز، به جذابیت فیلم افزوده است.

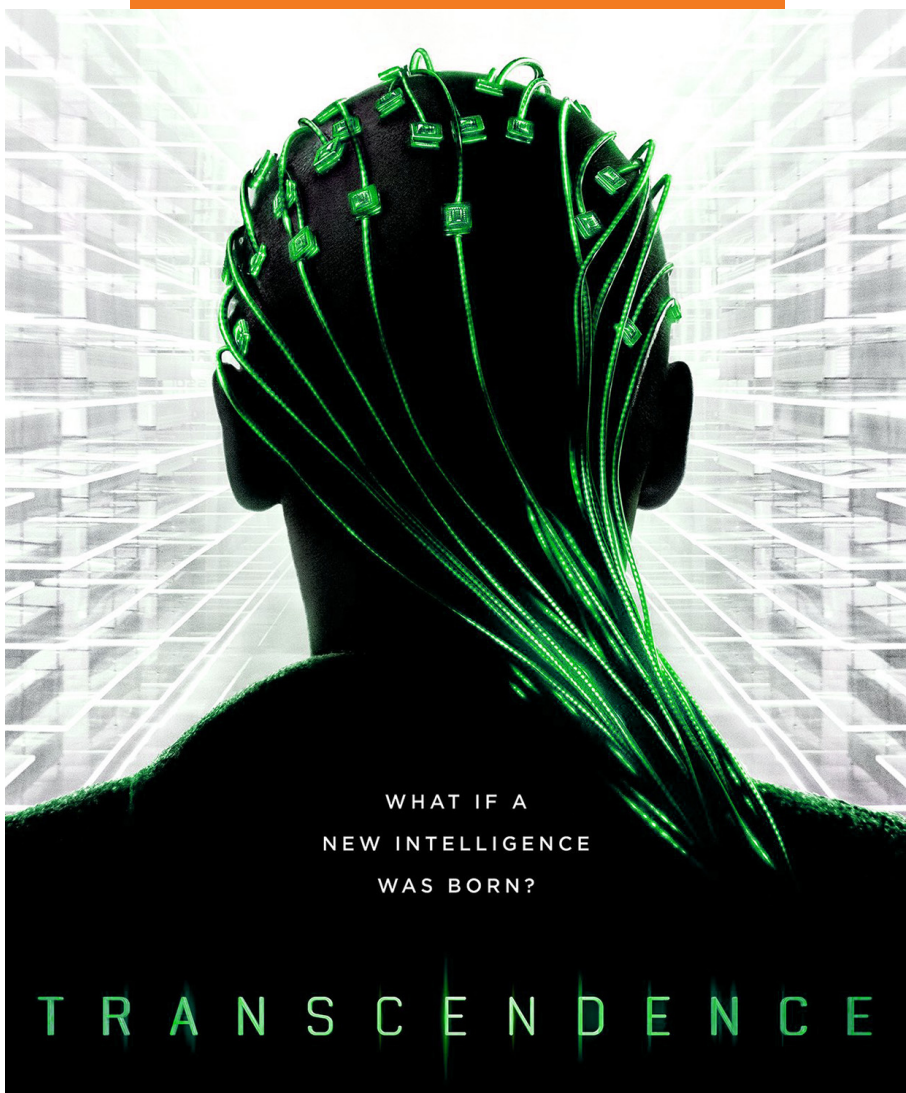
**جلوه‌های بصری:** فیلم از نظر بصری، از جذابیت بالایی برخوردار است. استفاده از رنگ‌های سرد و نورهای سفید، به خوبی حس علمی و اتمسفر سرد داستان را منتقل می‌کند. همچنین جلوه‌های ویژه دیجیتال، احساس تحول شخصیت‌ها را با دقت و ظرافت به نمایش می‌گذارند.

**مفاهیم فلسفی:** فیلم تعالی با طرح پرسش‌هایی بزرگ در مورد آگاهی، انسانیت و هوش مصنوعی، فضایی تأمل‌برانگیز ایجاد می‌کند که برای مخاطبان علاقه‌مند به مباحث علمی و فلسفی بسیار جذاب است.

## نکات منفی فیلم

**ریتم کند:** یکی از انتقادات وارد بر این فیلم، کند بودن روند داستان در میانه آن است. برخی از ایده‌های عمیق فیلم به‌طور کامل پرورش نمی‌یابند و این امر می‌تواند باعث خستگی مخاطب شود.

**پایان مبهم:** پایان فیلم به‌عمد باز و مبهم طراحی شده است و پرسش‌هایی را درباره‌ی خیر و شر در ذهن تماشاگر باقی می‌گذارد. این نوع پایان‌بندی ممکن است برای برخی از مخاطبان گیج‌کننده باشد.



## در باب بهترین بازی سال: این بار چه عنوانی در کالیفرنیا تاج‌گذاری خواهد کرد؟

**علیرضا کامرو**  
دانشجوی مهندسی کامپیوتر  
دانشکده فاریاب دانشگاه تهران  
alirezakamru@gmail.com



مراسم گیم اوارز را می‌توان معتبرترین مراسم اهدای جوایز در زمینه گیم دانست که هر ساله این صنعت و جامعه گیمرها را به تب و تاب می‌اندازد و عنوان‌های بزرگی را به چالش می‌کشد تا در نهایت بهترین بازی را در هر بخش مشخص کند. اما در میان همه این جوایز ارزشمند، تمام چشم‌ها به صحنه انتخاب بهترین بازی سال دوخته می‌شود. از طرفداران تا مخاطبان این مراسم و مخصوصاً بازی‌سازانی که یکی از اهداف بلندپروازانه‌ی خود را تصاحب این جایزه می‌دانند. هر دوره این جایزه به عنوان متفاوتی می‌رسد که این تفاوت می‌تواند از سبک تا فرهنگ را در برگیرد. امسال هم مانند سال‌های پیش تنوع در بازی‌های منتخب بسیار بالاست و ما در این بخش قصد داریم تا نامزدهای اصلی احتمالی این جایزه را معرفی کنیم. در ادامه با ما همراه باشید

**عرفان رزمانی مقدم**  
دانشجوی مهندسی کامپیوتر  
دانشکده فاریاب دانشگاه تهران  
erfan.ramezani@ut.ac.ir



**احمد محمدی**  
دانشجوی مهندسی کامپیوتر  
دانشکده فاریاب دانشگاه تهران  
mohammadi.ahmad@ut.ac.ir



۱۰ دقیقه

### HADES II

1

سازنده: Supergiant games

ناشر: Supergiant games

تاریخ انتشار: 25 سپتامبر 2025

پلتفرم‌ها: PC, Switch

سبک: rogue-lite/action adventure

نمره متاکریتیک: 100/95



منحصربه‌فرد در هر دور بازی و حدود شش سلاح متفاوت، نمی‌گذارد هیچ‌گاه از آن خسته شوید. بازی از منظر هنری نیز پیشرفت کرده است و در ارائه داستان هم گام‌های مثبت بزرگی برداشته است. افزون بر این، بازی در زمینه صوتی و موسیقی متن هم قدرتمند ظاهر شده است و درست مانند نسخه قبل و حتی بیشتر، در مبارزه با باس‌فایت‌ها شما را به وجد می‌آورد.

با در نظر گرفتن تمام این جوانب و علاقه مشترک مخاطبان و منتقدان به این عنوان، می‌توان گفت هیدیز ۲ به‌درستی در بین نامزدهای نهایی بهترین بازی سال قرار گرفته است.

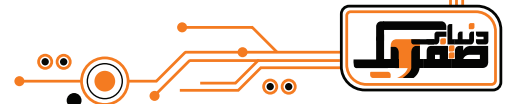
نسخه دوم می‌توانید دو مسیر جداگانه را در پیش بگیرید: یا به دنیای زیرین و خانه هیدیز برگردید و کروئوس را، که آن را به زور تصاحب کرده است، شکست دهید، یا مسیر بالا را پیش بگیرید و برای کمک به بستگان خود، به بالاترین نقاط کوه المپ صعود کنید. بازی در هر دو مسیر مبارزات جذاب و داستان‌گویی باکیفیتی را ارائه می‌دهد.

با وجود ایرادات کوچکی مانند افت در طراحی کاراکترهای فرعی و روند کلی داستان که به نسخه دوم هیدیز وارد است، این بازی تقریباً در هر جنبه‌ای که از نسخه اول فکرش را بکنید، چند درجه ارتقا پیدا کرده است. در زمینه گیم‌پلی نرم و روان‌تر به نظر می‌رسد و با داشتن ویژگی‌های

استودیوی سوپرگاینت گیمز پس از بازی هیدیز 1 و روایت داستان جذاب و فوق‌العاده شاهزاده زگروئوس، این بار بازگشته است تا گیمرها را با خواهر او، یعنی شاهدخت ملیئوئه، همراه کند. تصمیمی که در ابتدا به مذاق تمام طرفداران خوش نیامد؛ چرا که ایده‌ای تکراری به نظر می‌رسید. اما سازندگان نشان دادند که برای همراه کردن مخاطبان با خود، تبحر و مهارت کافی را دارند.

هیدیز ۲ بسیار بزرگ‌تر از نسخه اول ظاهر می‌شود و تقریباً دو برابر آن محتوای جذاب و سرگرم‌کننده ارائه می‌دهد. برخلاف هیدیز ۱ که باید دنیای زیرین را به مقصد سطح زمین ترک می‌کردید، در

شماره بیستم، آذر ماه ۱۴۰۴







## Donkey Kong Bananza

سازنده: Nintendo EPD

ناشر: Nintendo

تاریخ انتشار: 17 جولای 2025

پلتفرم: Nintendo Switch 2

سبک: platforming / action-adventure

نمره متاکریتیک: 100/91

# 2

تنوع مکانیک‌ها، به‌ویژه قابلیت تخریب و توانایی‌های ویژه دانکی کانگ در تبدیل شدن به موجودات دیگر، تجربه‌ای پویا و خلاقانه رقم می‌زند که حس ماجراجویی را تقویت می‌کند. با این حال، بازی خالی از ایراد نیست؛ مشکلات دوربین در برخی صحنه‌ها و افت فریم‌های گاه‌وبیگاه می‌توانند از روان بودن تجربه بازی بکاهند، هرچند تأثیر آن‌ها چندان جدی نیست.

در مجموع، دانکی کانگ بانانزا تجربه‌ای شاد، پرانرژی و سرشار از خلاقیت است که بار دیگر نشان می‌دهد نینتندو چگونه با حفظ هویت خود، همچنان می‌تواند گیمرها و منتقدان را شگفت‌زده کند.

فرمولی که سرگرمی را در مرکز توجه قرار می‌دهد. داستان بازی شاید عمیق‌ترین نقطه قوت آن نباشد، اما روایت بامزه و دوست‌داشتنی‌اش به‌خوبی مخاطب را با خود همراه می‌کند و پایانی رضایت‌بخش ارائه می‌دهد.

هسته اصلی گیم‌پلی بر پایه گشت‌وگذار در محیط‌ها، عبور از چالش‌ها و پیشروی به لایه‌های عمیق‌تر جهان بازی بنا شده است. هرچند ساختار کلی روندی نسبتاً خطی دارد، اما آزادی عمل قابل‌توجهی در اختیار بازیکن قرار می‌گیرد. شما می‌توانید با استفاده از سیستم تخریب‌پذیری چشمگیر، محیط را زیرورو کنید و آیتم‌های پنهان را بیابید یا با تمرکز بر مبارزات و عبور سریع از موانع، مسیر خود را کوتاه‌تر سازید.

در دنیای ویدئوگیم، نینتندو همواره نامی مترادف با خلاقیت، اصالت و سرگرمی ناب بوده است؛ شرکتی که از نخستین سال‌های فعالیت خود تا امروز، وفادار به اصولی مانده است که بازی را پیش از هر چیز، وسیله‌ای برای لذت و شگفتی می‌داند. همین نگرش خاص باعث شده است آثار این کمپانی، همواره حال‌وهوایی منحصربه‌فرد داشته باشند و مخاطبان را با هر تجربه تازه‌ای، به جهانی رنگارنگ و پرهیجان دعوت کنند.

دانکی کانگ بانانزا، جدیدترین نسخه از مجموعه کلاسیک دانکی کانگ، دقیقاً در همین مسیر قدم برمی‌دارد. این بازی بدون آن‌که به‌دنبال ساختار شکنی‌های عجیب باشد، بر همان فرمول موفق و آشنای نینتندو تکیه دارد؛

## Death Stranding 2: On The Beach

سازنده: Kojima Productions

ناشر: Sony Interactive Entertainment

تاریخ انتشار: 24 ژوئن 2025

پلتفرم: Playstation 5

سبک: Action adventure

نمره متاکریتیک: 100/89

# 3



همه‌چیز عمیق‌تر و تأثیرگذارتر شده است. کوچیما همچنین از انتقادهای وارده به نسخه اول درس گرفته و گیم‌پلی در این نسخه رشد چشمگیری داشته است. حرکت در محیط‌ها نرم‌تر و مأموریت‌ها هدف‌دارتر شده‌اند. همچنین تصمیم‌های بازیکنان واقعاً روی دنیای اطرافشان تأثیرگذار است. دیگر جابجایی بسته‌ها تنها نکته گیم پلی نیست و حالا هر

داریم: بزرگ‌تر، کامل‌تر و احساسی‌تر از قبل. از همان لحظه شروع، بازی حس ورود به دنیایی واقعاً زنده را القا می‌کند. گرافیک با موتور جدید دسیما واقعاً خیره‌کننده است؛ از ساحل‌های تار و مه‌گرفته تا کوه‌های پوشیده از برف، هر گوشه دنیا با جزئیاتی دیوانه‌کننده طراحی شده است. حس تنهایی، سکوت خاص و فضاهای وهم‌آلود هنوز از نسخه اول وجود دارند، ولی در نسخه دوم

وقتی نسخه اول این بازی منتشر شد، خیلی از گیمرها با خودش گفتند: «این دیگر چه جور بازی‌ای است؟» اما همان تجربه متفاوت و آرام توانست تبدیل به یکی از خاص‌ترین آثار نسل خود شود. حالا کوچیما بازگشته و با نسخه دوم کاری کرده است که حتی سخت‌گیرترین منتقدان هم آن را تحسین می‌کنند. Death Stranding 2 دقیقاً همان چیزیست که از یک دنباله واقعی انتظار

گیمرها مناسب نباشد، اما برای آن دسته که دنبال تجربه‌ای خاص، فلسفی و متفاوت هستند، یکی از بهترین بازی‌های امسال محسوب می‌شود. کوچیما نشان داده هنوز هم می‌تواند با فرمول‌های مخصوص خود بازی‌های کلاس جهانی بسازد و به همین دلیل، اصلاً عجیب نیست که نامش را در بین نامزدهای اصلی بهترین بازی سال ۲۰۲۵ ببینیم؛ زیرا Death Stranding 2 فقط یک بازی ساده نیست، بلکه یک تجربه واقعی است.

بازیگران نسخه قبلی مانند نورمن ریداس، لئا سیدو و حتی چهره‌های جدید مانند ال فایننگ فوق‌العاده ظاهر شده‌اند و روابط میان شخصیت‌ها کاملاً طبیعی و باورپذیر هستند. موسیقی نیز مانند نسخه قبل شاهکار لودویگ فورسل است؛ همیشه در زمان درست، با یک قطعه آرام یا پرتنش، ضربان قلب بازیکن را با ریتم بازی هماهنگ می‌کند. Death Stranding 2 شاید برای سلیقه همه

سفر پر از اتفاق، چالش و کشف جدید است. مبارزات هم بسیار پرتنش‌تر شده‌اند و ارتباط با شخصیت‌های دیگر نقش مهم‌تری در تجربه کلی بازی پیدا کرده است.

جایی که Death Stranding 2 واقعاً درخشان عمل می‌کند، بخش داستان بازی است. کوچیما این بار تمرکزش را بر روایت گذاشته و نتیجه آن داستانی سنگین، احساسی و گاهی دل‌خراش است که تا مدت‌ها در ذهن می‌ماند.



## Hollow Knight: Silksong

سازنده: Team Cherry

ناشر: Team Cherry

تاریخ انتشار: 4 سپتامبر 2025

پلتفرم‌ها: Pc, Switch, Xbox, Playstation

سبک: metroidvania/action platformer

نمره متاکریتیک: 100/91

# 4

تا طراحی هنری کم نظیر که هر فریم از بازی چشم‌های بیننده را مجذوب خود می‌کند. برای بخش موسیقی هم کریستوفر لارکین برگشته است تا بار دیگر گوش‌هایتان را نوازش کند. این بازی در حقیقت سبک خود یعنی سبک مترویدوینیا را به تعالی می‌رساند و به جرئت می‌توان گفت که جزو بهترین بازی‌های تاریخ در این سبک به حساب می‌آید.

سیلک سانگ بدون شک از برترین بازی‌های امسال است و باید دید که آیا پس از نامزد شدن در مراسم گیم اوردز که قطعی بنظر می‌رسد، می‌تواند این مقام مهم را به خود اختصاص دهد یا نه.

گیر و دشمنان چابک و قدرتمند بروید و از سد بخش‌های پلتفرمینگ بسیار چالشی آن عبور کنید تا بتوانید در نهایت به بالا ترین بخش‌های نقشه بازی صعود کنید و آن را به پایان برسانید. البته بازی یک بخش سوم مخفی شامل ادامه داستان و پایان حقیقی نیز دارد که به تمام بازیکنان به شدت پیشنهاد می‌شود آن را نیز از دست ندهند چرا که بدون آن، تجربه بازی عملاً ناقص می‌ماند.

سیلک سانگ در اکثریت بخش‌ها نسبت به هالونایت پیشرفت‌های چشم‌گیری داشته است. از گیم‌پلی فوق‌العاده بهتر، خلاقانه‌تر و گسترده‌تر گرفته

دنباله بازی هالونایت که در سال ۲۰۱۷ عرضه شده بود، پس از سال‌های طولانی انتظار بالاخره امسال منتشر شد. انتشاری که این اواخر و در سال‌های پایانی دیگر خون‌گیمرها را به جوش آورده بود ولی استودیو تیم چری ضرب‌المثل "گر صبر کنی ز غوره حلوا سازیم" را به خوبی معنا کرد و نشان داد که این انتظار، شدیداً ارزشش را داشته است.

شما در این بازی در نقش شخصیت نام‌آشنای نسخه اول یعنی هورنت قرار می‌گیرید تا با او و داستان جذابش در سرزمین جدید فارلوم همراه شوید، به مبارزه با باس فایت‌های جذاب و نفس

## kingdom come deliverance 2

سازنده: Warhorse Studios

ناشر: Deep Silver

تاریخ انتشار: 4 فوریه 2025

پلتفرم‌ها: Pc, Xbox, Playstation

سبک: Action-rpg

نمره متاکریتیک: 100/88

# 5



در بخش‌های خوب و قابل دفاع نسخه قبل مثل مذاکره کردن با مردم و نگهبانان یا جیب‌بری، تغییر عمده‌ای ایجاد نشده است، اما در برخی مکانیزم‌ها مانند مبارزه که آنچنان عامه‌پسند نبودند، از فرمول عمیق پیشین خود فاصله و کمی به مخاطبان آسان‌تر گرفته است. این تغییر را می‌توان یکی از مهم‌ترین دلایل محبوبیت بیشتر نسخه دوم نسبت به نسخه قبل و حضور آن در بین شش نامزد نهایی بازی سال دانست.

حال فقط باید منتظر ماند و دید که آیا پرندۀ بخت و اقبال در مراسم گیم آواردز بر شانۀ شاهزادۀ هنری و استودیو وارهورس خواهد نشست یا خیر.

جانبی بازی نیز خارق‌العاده عمل می‌کنند و مشخصاً هرکدام به‌دقت نوشته و اجرا شده‌اند. این ماموریت‌ها چنان از نظر محتوایی قوی هستند که ممکن است برایتان حتی تبدیل به محتوای اصلی بازی شوند و در جریان تجربه آن‌ها اصلاً احساس نمی‌کنید در حال انجام کاری هستید که تأثیری در داستان اصلی ندارد. همه ماموریت‌ها دارای روایتی مشخص و منسجم به‌همراه پیش‌های داستانی هستند که باعث می‌شود به‌مراتب جذاب‌تر و به‌یادماندنی‌تر از ماموریت‌های جانبی هر بازی دیگری باشند.

بازی در این نسخه از نظر گیم‌پلی نیز تغییرات مثبتی داشته است و با وجود اینکه

همان‌طور که احتمالاً تا کنون متوجه شده‌اید، امسال سال نسخه دوم بازی‌هایی بوده که با نسخه اولشان محبوبیت قابل قبولی کسب کرده‌اند و از حق نگذریم این دنباله‌ها هم هیچکدام مخاطبان را ناامید نکرده‌اند و Kingdom Come: Deliverance 2 هم از این قاعده مستثنی نیست.

داستان بازی دوباره ما را با شاهزادۀ هنری همراه می‌کند که در ببحوحۀ جنگی که در کشورش، بوهیمیا، در حال وقوع است سعی در گرفتن انتقام خانواده‌اش از پادشاه مجارستانی، یعنی سیگیسموند، دارد.

اما علاوه بر داستان اصلی، ماموریت‌های



## Clair Obscur: Expedition 33

سازنده: sandfall interactive

ناشر: kepler interactive

تاریخ انتشار: 24 آوریل 2025

پلتفرم‌ها: Pc, Xbox, Playstation

سبک: turn-based rpg/adventure

نمره متاکریتیک: 100/92

# 6

کرده است و با داستان و پیش‌هایش تک‌تک احساسات مخاطب را به چالش می‌کشد. شخصیت‌ها همگی خوب ساخته و پرداخته شده‌اند و هیچ‌کدام احساس اضافی یا پوچ بودن نمی‌دهند؛ این موضوع هم‌زادپنداری و درک آن‌ها را آسان‌تر می‌کند.

همۀ این‌ها، در کنار گیم‌پلی نوآورانه و ترکیبی، صدآپیشگان درجه یک و دیالوگ‌های تأثیرگذار و به‌یادماندنی، باعث می‌شود که Expedition 33 به سادگی در جمع نامزدهای بهترین بازی سال جای بگیرد.

که تا ریزترین جزئیاتشان نیز با دقتی مثال‌زدنی کشیده و رنگ‌آمیزی شده‌اند. این امر در کنار گرافیک بالا و چشم‌نواز باعث می‌شود این دنیا را ساعت‌ها بدون خسته شدن یا احساس تکرار کاوش کنید. البته نمی‌توان از نقش مهم و حیاتی موسیقی در این مسیر نیز چشم‌پوشی کرد. بازی برای هر منطقه و دشمن، موسیقی منحصربه‌فردی را ارائه می‌دهد که همین امر لذت مبارزات و گشت‌وگذار در دنیای بازی را چندین برابر می‌کند. این اثر از نظر روایت هم بسیار خوب عمل

پیش از عرضه، احتمالاً کمتر کسی پیدا می‌شد که اسم این عنوان حتی به گوشش خورده باشد، اما اکنون هیچ‌کس از حضور آن در بین نامزدهای اصلی بهترین بازی سال تعجب نمی‌کند.

Expedition 33 به معنای واقعی کلمه یک عنوان کامل است و در تمام جنبه‌هایی که از یک بازی انتظار دارید، کم‌نقص عمل می‌کند؛ تقریباً همه چیز در آن نقطه قوت محسوب می‌شود.

هر کدام از مناطق دنیای بازی از لحاظ بصری همانند تابلوهای نقاشی متفاوت و زنده‌ای هستند

## بازی های افتخاری:

امسال، فراموش شدند و یا در حاشیه قرار گرفتند. اما این بازی‌ها به اندازه‌ای دل مخاطبان را برده‌اند که ما را مجاب به اضافه کردن بخشی برای ادای احترام به آن‌ها کرده‌اند تا شما هم از تجربه این بازی‌های جذاب غافل نشوید:

امسال در کنار بازی‌هایی که به عنوان نامزدهای اصلی این جایزه معرفی کردیم، بازی‌هایی نیز حضور داشتند که شاید به دلیل منتقدپسند نبودن یا نداشتن توان و یرای رقابت با بازی‌های مهم‌تر

## blue prince

سازنده: Dogubomb

ناشر: Raw Fury

تاریخ انتشار: 10 آوریل 2025

پلتفرم‌ها: Pc, Xbox, Playstation

سبک: puzzle adventure/roguelike

نمره متاکریتیک: 100/92



## Ghost of Yōtei

سازنده: Sucker Punch Productions

ناشر: Sony Interactive Entertainment

تاریخ انتشار: 2 اکتبر 2025

پلتفرم‌ها: Play station 5

سبک: اکشن / ماجراجویی

نمره متاکریتیک: 100/90



## silent hill F

سازنده: NeoBards entertainment

ناشر: Konami

تاریخ انتشار: 25 سپتامبر 2025

پلتفرم‌ها: Pc, Playstation 5, Xbox

سبک: Survival horror

نمره متاکریتیک: 100/86



## Split Fiction

سازنده: hazelight studios

ناشر: Electronic Arts

تاریخ انتشار: 6 مارچ 2025

پلتفرم‌ها: Pc, Xbox, Playstation

سبک: co-op/action adventure

نمره متاکریتیک: 100/91



با تمام این اوصاف می‌توان گفت که امسال از نظر بازی خوب و تاثیرگذار، جزو سال‌های کم‌نظیر این مراسم بوده است و انتظارات مخاطبان را از این مراسم بسیار بالا برده است. مراسم گیم‌آوردز امسال در تاریخ ۱۱ دسامبر (۲۰ آذر) برگزار می‌شود و بالاخره مشخص می‌شود که چه عنوانی امسال تاج بهترین بازی سال را روی سر خواهد گذاشت.

## رضا قمرالاسلا

دانشجوی مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
rezaqomyasl@gmail.com



## هفت خان توسعه بک اند خان صفرم و خان اول: سفری به قلمرو بک اند

۱۲ دقیقه

### قسمت چهارم: رسیدن به مقصد

**خان ششم:** فرماندهی لشکر سرورها (ابزارهای DevOps، استقرار کد و مانیتورینگ سیستم)

**خان هفتم:** بازار کار و گنجینه مهارت‌ها (رزومه، مصاحبه و مهارت‌های نرم لازم برای موفقیت)

اما خب، قبل از اینکه بزنیم به دل جاده و وارد خان صفرم بشیم، می‌خوایم بگیم که بک‌اند اصلاً چیه و چرا تا این حد برای هسته‌ی هر نرم‌افزاری اهمیت داره.

### خان صفرم: ورود به قلمرو پنهان

#### (بک‌اند چیست و چرا مهم است؟)

این خان، دروازه ورود به سفر است؛ جایی که با مفاهیم بنیادی دنیای سرور آشنا می‌شویم و ابزارهای اولیه خود را برمی‌گزینیم.

### پرده‌برداری از قلمرو پنهان (بک‌اند چیه؟)

#### (ماجرای پشت پرده نرم‌افزارها)

فرض کنید که می‌خواید از نرم‌افزاری مثل آمازون، دیجی‌کالا یا هر نرم‌افزار دیگری که توی وب هست، استفاده کنید. وقتی وارد سایت می‌شید، به صفحه می‌بینید که داخلش یک سری عکس، منو، دکمه و این‌ها وجود داره که شما می‌تونید توی سایت باهاشون بگردید و سایت رو بالا و پایین کنید. این بخشیه که شما به عنوان کاربر مستقیماً باهاش سروکله می‌زنید. ما به اون بخش می‌گیم Client-Side یا همون فرانت‌اند؛ یعنی جایی که کاربر باهاش کار می‌کنه و تمام اتفاقات بصری اونجا رخ می‌ده. کدهای فرانت‌اند توی مرورگر شما اجرا می‌شن و کارشون اینه که داده‌ها رو خوشگل کنن و نشون بدن. البته حواستون باشه فرانت فقط مرورگر نیست؛ می‌تونه نرم‌افزاری باشه که مستقیم روی گوشیتون نصب می‌کنید یا حتی دستگاه عابر بانکی باشه که ازش پول می‌گیرید.

خب، جدای از این بخش (فرانت‌اند)، به قسمت دیگه هم داریم که تمام کار فکری و محاسباتی رو انجام می‌ده. مثلاً فرانت‌اند لیست محصولات سایت، اینکه امتیازش چقدره، چقدر موجودی داره، قیمتش چنده، تخفیف داره یا نداره رو از یه جایی می‌گیره. این اطلاعات نه تنها باید به جا ذخیره بشن، بلکه باید قبل از نمایش، منطق‌های پیچیده‌ای هم روشون اعمال بشه؛ مثلاً بررسی بشه که آیا کاربر اصلاً اجازه دیدن این قیمت رو داره یا نه، یا اگه به تخفیف خاص خورده، محاسبه قیمتش چطوره. تمام این کارها باید سمت یه سرور هندل بشن که به این بخش ما می‌گیم Server-Side یا همون بک‌اند خودمون.

سلام و درود بی‌کران بر شما مسافران جاده دانش!

توی این مقاله و سری‌های بعدی، می‌خوایم ببینیم راه و رسم «بک‌اندی شدن» چیه و چطوره. این سفر، مسیر کسایه که می‌خوان مغز متفکر و موتور محرک یه نرم‌افزار باشن.

برای اینکه توی این مسیر طولانی و پرماجرا گم نشیم، یک نقشه‌راه جامع و مرحله‌به‌مرحله با عنوان «هفت خان» آماده کردیم. قراره قدم به قدم از صفر شروع کنیم، بریم جلو و از مفاهیم اولیه گرفته تا مقیاس‌پذیری و بازار کار رو فتح کنیم. این مسیر توی چهار قسمت مجزا (چهار مقاله) با شما همراه خواهد بود:

## نقشه‌راه

## هفت خان

## توسعه‌دهنده بک‌اند



### قسمت اول (همین مقاله): دروازه‌ی ورود

**خان صفرم:** ورود به قلمرو پنهان (مفاهیم پایه‌ای که برای شروع این سفر بهشون نیاز داریم)

**خان اول:** آماده‌سازی برای نبرد (فریم‌ورک‌ها، دیتابیس رابطه‌ای و ابزارهای ضروری)

### قسمت دوم: عمق و استحکام

**خان دوم:** غواصی در اقیانوس داده‌ها (تسلط عمیق بر پایگاه‌های داده، NoSQL و Caching)

**خان سوم:** نگهبانان و قفل‌های آهنین (اصول امنیت، احراز هویت و طراحی API های استاندارد)

### قسمت سوم: مهندسی یک سیستم بزرگ

**خان چهارم:** معماران بناهای ماندگار (معماری نرم‌افزار، اصول SOLID و دیزاین پترن‌ها)

**خان پنجم:** رام کردن هیولای مقیاس (میکروسرویس‌ها، صف‌ها و مدیریت ترافیک‌های بالا)

بک‌اند در واقع سه وظیفه خیلی مهم داره:

**مدیریت داده:** ذخیره، سازماندهی و بازیابی تمام اطلاعات حیاتی مثل رمز عبور کاربران (البته به صورت هش شده)، لیست سفارشات، مشخصات محصولات و...

**اجرای منطق کسب‌وکار (Business Logic):** هسته اصلی نرم‌افزار، یعنی قوانین و فرآیندهای کسب‌وکار (مثلاً اینکه چطوری پول از حساب کاربر کم بشه، یا چطور یک سفارش تأیید بشه).

**تأمین امنیت:** مطمئن شدن از اینکه هر کسی فقط به اطلاعاتی دسترسی پیدا کنه که حقشه و جلوگیری از حمله‌های مختلف سایبری.

حالا که فرق این دو دنیا رو حسابی فهمیدیم، بریم و سیر و سلوک بک‌اندی شدن رو با هم شروع کنیم.

## آموختن زبان ارتباط (آشنایی با شبکه)

### آشنایی با زبان مشترک اینترنت

شبکه، اولین چیزیه که باید در موردش بدونیم. دلیلش هم واضحه: بک‌اند کلاً روی شبکه‌ست و باید زبان ارتباطات رو بدونه.

البته قرار نیست توی بک‌اند یه شبکه کار حرفه‌ای بشید (مثل متخصصانی که کانفیگ روتر و سوئیچ انجام می‌دن). فقط لازمه که اولش یه دانش پایه محکم از شبکه داشته باشید که اگه بعداً به مشکلی چیزی برخوردید، مثلاً یه درخواست تایم‌آوت شد، بتونید گلیم خودتون رو از آب بکشید بیرون. این دانش به شما این دید رو می‌ده که بفهمید داده‌ها چطور توی اینترنت جابه‌جا می‌شن.

برای شبکه، اگه در این حد بدونید که IP چیه (که مثل آدرس منحصربه‌فرد دستگاه شما توی شبکه‌ست)، Port چیه (که مثل شماره ورودی یک سرویس خاص روی اون آدرسه)، DNS چطوری اسم سایت‌ها رو به همون آدرس‌های IP ترجمه می‌کنه، فرق UDP با TCP چیه و مفاهیم پایه این شکلی، کافیه.

## انتخاب قلب تپنده (زبان برنامه‌نویسی)

### یادگیری فن کدنویسی و اصول مهندسی

بعد از اینکه یکم دستتون تو شبکه گرم شد و اصولش رو فهمیدید، اینجا نوبت انتخاب یه زبان برنامه‌نویسیه که باهش کدتون رو پیاده‌سازی کنید.

زبان‌های برنامه‌نویسی زیادی سمت بک‌اند وجود دارن و استفاده می‌شن. هرکدوم یه عالمه مزایا و معایب دارن که بسته به نوع پروژه باید یکی رو انتخاب کرد. مثلاً Ruby، Java، C#، PHP، Node.js، Go، Python و تعداد فراوانی از زبان‌های برنامه‌نویسی دیگه، در این میدان حضور دارن.

نکته مهم اینه که: شما باید یکیش رو انتخاب کنید.

اگه از قبل هرکدوم از این زبان‌ها رو بلدید یا باهش کار کردید، می‌تونید همون رو انتخاب کنید. این خیلی خوبه که از نقطه‌ای شروع

کنید که توش قوی هستید. اما اگه تا الان با هیچ‌کدوم از زبان‌های برنامه‌نویسی کار نکردید، پیشنهاد من اینه که با Python شروع کنید و مفاهیمش رو یاد بگیرید. پایتون به خاطر خوانایی بالا و سادگی، شما رو درگیر پیچیدگی‌های زبانی نمی‌کنه و می‌تونید سریع‌تر مفاهیم اصلی بک‌اند رو یاد بگیرید.



فارغ از اینکه چه زبانی رو انتخاب می‌کنید، باید روی مفاهیم اصلی اون زبان مسلط بشید:

**ساختارهای داده (Data Structures):** اینکه لیست، دیکشنری، آرایه، درخت یا گراف چیه و هرکدوم کجا به کار میان.

**الگوریتم‌ها (Algorithms):** چطوری کدی بنویسیم که کارآمد باشه و سرعتش بالا باشه (مثلاً چطوری یه لیست رو جستجو یا مرتب کنیم).

**مفاهیم شیء‌گرایی (OOP):** اینکه کلاس و شیء و وراثت چیه و چطوری کدمون رو تمیز و ماژولار بنویسیم.

**همزمانی (Concurrency & Multi-threading):** این مفهوم شاید کمی پیچیده باشه اما حیاتیه. بک‌اند برخلاف کدهای معمولی که خط به خط اجرا می‌شن، باید بتونه همزمان به صدها یا هزاران کاربر جواب بده. باید یاد بگیرید چطور برنامه‌ای بنویسید که وقتی یک کاربر منتظر جواب دیتابیس، برنامه متوقف نشه و بتونه کار بقیه کاربرها رو راه بندازه (مفهوم Blocking vs Non-Blocking).

این مفاهیم، هسته کار هر مهندس نرم‌افزار رو تشکیل می‌دن، نه فقط بک‌اند دولوپر.

## زبان مشترک بک‌اند و فرانت‌اند (پروتکل HTTP)

### زبان ارتباطی بک‌اند و فرانت‌اند

بعد از انتخاب زبان برنامه‌نویسی، باید برید و یکم در مورد HTTP (Hypertext Transfer Protocol) بدونید. گفتیم که این پروتکل، زبان گفتگوی وب و اساس کار بک‌اند ماست. شما باید بفهمید که سرور شما چطوری درخواست‌ها رو دریافت می‌کنه و بهشون پاسخ می‌ده.

این درک شامل:

**متدها (Methods):** چه متدهایی داره؟ مثلاً GET (برای دریافت داده)، POST (برای ارسال و ساختن داده)، PATCH و PUT (برای ویرایش داده) و DELETE (برای پاک کردن داده). هرکدام از اینها باید دقیقاً برای کاری که تعریف شدن، استفاده بشن.

**هدر (Header):** هدر در واقع اطلاعات اضافی و متادادهای درخواست رو حمل می‌کنه؛ مثل نوع محتوایی که فرستاده می‌شه (مثلاً JSON، توکن‌های امنیتی یا کوکی‌ها).

**بادی (Body):** همون محتوای اصلی داده‌ای که فرستاده یا دریافت می‌شه. مثلاً وقتی کاربر توی یه فرم ثبت نام می‌کنه، اطلاعاتش توی بادی درخواست POST قرار می‌گیره.

**کدهای وضعیت (Status Codes):** باید بدونید که کدهای وضعیت مثل ۲۰۰ (موفقیت‌آمیز)، ۴۰۴ (پیدا نشد)، ۴۰۱ (مجوز نداری) و ۵۰۰ (خطای سرور) چیا هستن و کی باید از کدوم استفاده کرد. استفاده درست از این کدها، به فرانت‌اند و کلاینت کمک می‌کنه که بفهمه عملیات شما موفق بوده یا شکست خورده و دلیل شکست چی بوده.

**کوکی (Cookie):** کوکی‌ها تکه‌های کوچکی از داده هستن که سرور برای ذخیره وضعیت کاربر (مثلاً اینکه کی وارد شده) در مرورگر کلاینت نگه می‌داره.

درک این ساختار پروتکل، به شما کمک می‌کنه که بتونید باگ‌ها رو بهتر پیدا کنید و API‌های (رابط‌های برنامه‌نویسی) تمیزتری طراحی کنید.

البته این رو هم در نظر داشته باشید که HTTP تنها پروتکلی نیست که نیازه؛ پروتکل‌های دیگه‌ای هم هستن که هر کدوم در جای خودشون کاربرد دارن و در قسمت‌های بعدی معرفی‌شون می‌کنیم.

## خان اول: آماده‌سازی برای نبرد

در این خان، با ابزارهای قدرتمندی که برای ساخت، سازماندهی و نگهداری نرم‌افزار لازم است، آشنا می‌شویم.

### زره و سلاح نبرد (فریم‌ورک‌ها و لایبرری‌ها)

#### (استفاده از فریم‌ورک‌های توسعه سریع)

بعد از اون، باید برید سراغ یک فریم‌ورک یا لایبرری داخل همون زبان برنامه‌نویسی که انتخاب کردید.

در واقع، اون فریم‌ورک یا لایبرری یه لایه میانجی هست بین اون زبان برنامه‌نویسی که انتخاب کردید و



پروتکل HTTP. این ابزارها، کارهای تکراری و پیچیده رو قبلاً برای شما نوشتن؛ مثلاً سیستم مسیریابی درخواست‌ها، مدیریت خطاها یا امنیت اولیه. شما با استفاده از این ابزارها می‌تونید سریع‌تر بک‌اند خودتون رو توسعه بدید و تمرکزتون رو بذارید روی منطق اصلی کسب‌وکار.

زبان‌های برنامه‌نویسی مختلف، فریم‌ورک‌ها و لایبرری‌های متفاوتی دارن:

**پایتون:** FastAPI (خیلی سریع و مدرن)، Flask (ساده و مینیمال)، Django (کامل و جامع)

**Go:** Gin, Echo, Fiber

**Node.js:** Express, NestJS

**نکته کلیدی:** نکته مهمی که وجود داره و باید حسابی تو ذهنتون باشه، اینه که: شما نباید سعی بکنید یک فریم‌ورک بشوید. یعنی برای مثال جنگو رو یاد بگیرید و اسم خودتون رو بذارید "جنگو دولوپر" و فقط کد جنگو بزنیند. شما باید به یه مهندس نرم‌افزار (Software Engineer) تبدیل بشید؛ یعنی کسی که به مفاهیم تسلط داره و می‌تونه با ابزارهای موجود، یک نرم‌افزار رو توسعه بده، نه کسی که فقط یک ابزار رو بلده. اگر فردا روزی پروژه شما نیاز به زبان یا فریم‌ورک دیگه‌ای داشت، باید توانایی این رو داشته باشید که اون ابزار رو یاد بگیرید و باهاش کار کنید. پس روی ابزارتون تعصب نداشته باشید و همیشه ذهنتون رو برای یادگیری ابزارهای جدید باز نگه دارید.

## خزانه‌داری اسناد (دیتابیس‌ها و SQL)

### (مدیریت داده‌های نرم‌افزار)

هر نرم‌افزار بک‌اندی نیاز به یک‌سری اطلاعات داره که باید یه جا ذخیره بشن؛ مثلاً لیست اسامی یوزرها با هاش پسوردشون، تاریخی که عضو سایت شدن، لیست محصولات یا موجودی کیف پول کاربران. این اطلاعات قراره توسط یه دیتابیس (Database) ذخیره و مدیریت بشن.

ما انواع مختلفی از دیتابیس‌ها رو داریم، ولی پرکاربردترین اون‌ها، دیتابیس‌های رابطه‌ای (Relational Databases) هستن. در این مدل، شما اطلاعات رو توی یه جدول ذخیره می‌کنید. مثلاً یه جدول داریم به اسم users که یک‌سری ستون داره (مثلاً name, email, created\_at و ...) و هر یوزر یه ردیف جدید داخل این ستون داره. این جداول با منطق مشخصی به هم ارتباط دارن. این نوع دیتابیس برای داده‌هایی که ساختار مشخص و روابط ثابتی دارن (مثل تراکنش‌های مالی یا مشخصات کاربر)، فوق‌العاده مطمئن و مناسبه.

یه زبان برنامه‌نویسی وجود داره به اسم SQL (Structured Query Language) که در واقع شما می‌تونید با استفاده از اون با دیتابیس‌های رابطه‌ای کار بکنید؛ یعنی توش جدول جدید بسازید، داخلش جستجو کنید، ردیف بسازید، پاک کنید یا ویرایش کنید. نیاز هست که هر بک‌اند دولوپری این زبان رو بلد باشه و توانایی داشته باشه که کوئری‌های بهینه بنویسه (کوئری‌هایی که دیتابیس رو گند نکنن).

**ابزارهای شیء‌گرایی (ORM):** همچنین نیازه که به یک لایبرری یا ORM (Object-Relational Mapping) که توی زبان برنامه‌نویسی شما وجود داره، مسلط بشید. ORM ها به شما اجازه می‌دن که با مدل‌های شیء‌گرایی زبان خودتون (مثلاً یک کلاس Python) با دیتابیس صحبت کنید، بدون اینکه مجبور باشید همه‌جا کوئری SQL خام بنویسید. این کار هم گد شما رو تمیزتر می‌کنه و هم از لحاظ امنیتی خیلی بهتره.

**نکته‌ای برای آینده دیتابیس‌ها:** البته ما دیتابیس‌های دیگه‌ای هم داریم که بهشون می‌گیم NoSQL. این دیتابیس‌ها ساختار رابطه‌ای ندارند و برای انواع خاصی از داده‌ها (مثل لاگ‌ها یا داده‌هایی که ساختار متغیری دارن) بهتر عمل می‌کنن. ما در مقالات بعدی، مفصل در مورد NoSQL و انواع مختلف اون (مثل MongoDB) صحبت می‌کنیم و می‌گیم که کی باید سراغ این مدل‌ها بریم. فعلاً تمرکزمون روی یادگیری عمقی دیتابیس‌های رابطه‌ای باشه.

## نقشه‌های جا به جایی: مهاجرت‌ها (Migrations)

### تغییرات دیتابیس و نگهداری ساختار (کد)

نرم‌افزارها عموماً تغییرات زیادی دارن. یعنی با گذر زمان، قابلیت‌های جدید اضافه و قدیمی‌ها ویرایش می‌شن. این تغییرات شامل تیل‌های دیتابیس شما هم هستن. مثلاً ممکنه امروز نیاز باشه به ستون جدید به جدول کاربرا اضافه کنیم.

اینجاست که بحث Migrations (مهاجرت‌ها) وسط می‌آد. مهاجرت‌ها سازوکارهایی هستن که به شما اجازه می‌دن تغییرات ساختاری دیتابیس رو (مثل اضافه کردن یک ستون یا تغییر نام جدول) به صورت کدی و قابل ردیابی مدیریت کنید. این کار تضمین می‌کنه که دیتابیس شما توی محیط توسعه، تست و نهایی، همیشه ساختار یکسانی داشته باشه و بتونید به راحتی تغییرات رو اعمال کنید و در صورت نیاز برگردونید. این مفهوم به شدت برای کار تیمی و استقرار در سرور مهمه.

برای مایگرشن ابزارهای متفاوتی وجود که معمولاً به صورت پیش فرض روی همون فریم‌ورکی که باهاش کار می‌کنید وجود دارن مثلاً Django یا Sequelize توی Node.js. اما لایبرری‌ها مستقلی هم وجود دارن مثل Alembic توی پایتون یا Flyway/Liquibase که برای مدیریت این تغییرات استفاده میشن.

## ابزارهای سپاه نرم‌افزار (ابزارهای ضروری)

### ابزارهای ضروری برای توسعه‌دهنده

علاوه بر یادگیری مفاهیمی که دقیقاً مرتبط با نوشتن یک کد بک‌اند هست، شما باید یک سری ابزار رو هم بلد باشید که بتونید با استفاده از اون‌ها، کد خودتون رو توسعه بدید، نگهداری کنید و به بهترین شکل ممکن به سرور بفرستید.

### الف. گیت (Git) برای کنترل نسخه

یکی از مهم‌ترین ابزارهایی که تقریباً باید همه برنامه‌نویس‌ها بلد باشن

و بتونن باهاش کار کنن، یک ورژن کنترل هست که معروف‌ترین اون‌ها Git هست. گیت مثل یه ماشین زمان برای کد شماست.

باید بدونید مفاهیمی مثل Commit (ثبت به بسته تغییرات)، Branch (انشعاب دادن برای کار روی یه ویژگی جدید)، Merge Request یا Pull Request (درخواست ادغام کد) و Merge (ادغام نهایی) چیه و چطوری باید با گیت کار کنید. تسلط بر گیت، همکاری شما رو توی تیم‌های نرم‌افزاری منظم و حرفه‌ای می‌کنه و اجازه نمی‌ده که کد یک نفر، کار بقیه رو خراب کنه. استفاده از پلتفرم‌هایی مثل GitHub و GitLab که برای مدیریت مخازن گیت استفاده می‌شن، توی این مرحله خیلی مهمه.

### ب. داکر (Docker) برای ایزوله‌سازی محیط

بعد از اون، یکی از ابزارهایی که یک بک‌اند دولوپر مدرن باید بدونه، داکر (Docker) هست. احتمالاً شما هم این جمله رو شنیدید که برنامه‌نویس‌ها میگن: «روی سیستم من کار می‌کرد!»

داکر اومده که این مشکل رو حل کنه. داکر به شما یه محیط ایزوله می‌ده که نرم‌افزار شما با تمام وابستگی‌ها، کتابخونه‌ها و محیط عملیاتی که نیاز داره، داخل اونجا اجرا بشه. این یعنی شما مطمئن می‌شید که کدی که روی لپ‌تاپ خودتون می‌نویسید، دقیقاً همون‌طوری که انتظار می‌ره، روی سرور یا سیستم همکارتون هم کار می‌کنه.

باید بدونید که Image (تصویر) داکر چیه و چطوری از روی اون Container (کانتینر) ساخته می‌شه و این چطوری به ما کمک می‌کنه که کدمون رو توی هر محیطی، بدون دردسر زیاد استقرار (Deploy) بدیم.

**نکته‌ای برای آینده:** در آینده‌ای نزدیک، وقتی پروژه‌های ما بزرگ‌تر بشن، ممکنه مجبور بشیم ده‌ها یا صدها داکر کانتینر رو همزمان مدیریت کنیم. اینجاست که ابزارهایی مثل Kubernetes برای هماهنگی کانتینرها وارد میدن می‌شن، که بحث اونا رو هم توی قسمت‌های پیشرفته‌تر دنبال خواهیم کرد.

### ج. تست و مستندسازی API

شما وقتی کد بک‌اند رو می‌نویسید، هنوز فرانت‌اندی وجود نداره که از طریق اون دکمه‌ها رو بزنیید و ببینید کتون کار می‌کنه یا نه. پس چطوری باید تست کنید؟

اینجاست که ابزارهایی مثل Postman (یا جایگزین‌هایی مثل Insomnia) وارد می‌شن. این ابزارها نقش «فرانت‌اند مجازی» رو بازی می‌کنن و به شما اجازه می‌دن درخواست‌های HTTP (مثل Login کردن) رو به سرور بفرستید و جواب رو ببینید.

و ابزار مهم دیگه Swagger (یا OpenAPI) هست. وقتی شما یه API می‌سازید، برنامه‌نویس فرانت‌اند از کجا باید بدونه چه آدرسی رو صدا بزنه و چه دیتایی بفرسته؟ شما نباید این‌ها رو شفاهاً توضیح بدید. ابزارهایی مثل Swagger به صورت خودکار یک صفحه مستندات (Documentation) تمیز از روی کد شما می‌سازن که همه چیز رو توضیح میده. این یعنی زبان مشترک و مستند بین شما و تیم فرانت.





**ماهان قدیرخانی**  
دانشجوی مهندسی کامپیوتر  
دانشکده فابریک دانشگاه تهران  
mahan.gk1384@gmail.com



## جعبه ابزار مهندسی

۲ دقیقه

### 1. Screeps:

این یکی کار هرکسی نیست! فکر کن یه بازی آنلاین دربرابر بازیکن‌های واقعی، جایی که سربازها فقط با کدهایی که تو نوشتی حرکت می‌کنن. حتی وقتی آنلاین میشی هم حیات اون‌ها ادامه پیدا می‌کنه و کدهات دارن اجرا میشن.



جاوا اسکریپت و هر زبان دیگه‌ای که بلدی رو بیار وسط تا تو این رقابت عظیم از رقبا شکست نخوری.

<screeps>



با عنایتِ دیر، رفاقتِ سردیر، مرحمتِ خدایِ کبیر و به همتِ شوری نفس گیر، بخش زیر برشته تعمیر درآمده تا فرصتی برای آشنایی بیشتر با جهان کنونی کامپیوتر توأمان با شیرینی فراهم گردد.

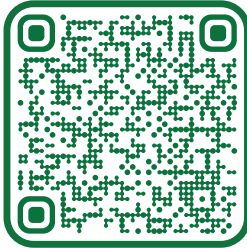
خلاصه که آقا ما توی این صفحه جا خوش کردیم تا با هم بریم و چندتا از چیزهای مختلفی که می‌تونن در روند زندگی روزمره ما تاثیر مثبتی بذارن و اون قالب خشکی که دور خودمون کشیدیم رو بشکنن، آشنا بشیم.

البته که دوستن‌شون به تنهایی کافی نیست و برنده اونیه که ازشون بهره ببره.

پس بریم که حسابی بترکونیم!

## 5. CS50:

اینجا می‌خواهم بدون ویزا و اپلای  
بفرستمت هاروارد!



این دوره از اونجایی که پیش نیاز  
خاصی نداره، به شدت به بچه‌هایی  
که ابتدای مسیر یادگیری هستند  
توصیه میشه. اما شمایی که بلدکار هستی هم بد نیست یه  
مدرک هاروارد بزاری تو رزومت (:

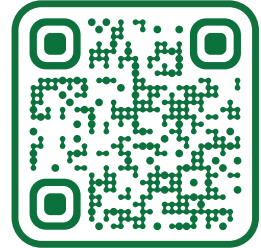
پ.ن 1: قابل ذکره که گروه‌های داخلی زیادی با دریافت مبلغی  
این دوره رو به زبان فارسی به شما آموزش میدن، ولی خودت رو  
دست کم نگیر و این دوره رو به انگلیسی و رایگان از منبع اصلی  
ببین .

پ.ن 2: اگه مدرک دهن پر کن‌تری خواستی یه سر به Coursera  
و Edx بزن.



## 2. Kaggle:

فکر کردی این همه دیتاست  
ماشین لرنینگ از کجا میاد؟  
بله جانم، کگل زحمتشو می‌کشه.  
یه ملت اونجا هرچی دیتاست بوده رو  
جمع کردن و گذاشتن که بهره‌بری.

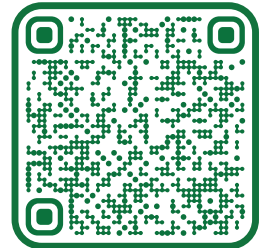


آمار وحشتناک 500+ هزار دیتاست شوخی نیست! تازه کلی  
مسابقه و چیزای دیگه هم داره.

# kaggle

## 3. Envato Tuts+ Illustrator:

با کمی تلطیف فضا (و کاملاً به  
خاطر سلیقه خودم)، بریم سراغ یه  
دوره تر و تمیز و جمع‌وجور یوتیوبی  
برای شروع کار با ایلاستریتور. دن  
آقای اسکات میاد و هرچی که نیاز  
داری تا با دنیای جذاب vectorها



آشنا بشی رو بهت میگه و با چندتا تمرین، جادوی گرافیک رو  
نشونت میده.



## 6. Codewars:

به دوجوی قرمز خوش آمدید.  
اینجا با انجام kata(چالش)های  
مختلف کد نویسی رنگ کمربندت  
تغییر می‌کنه و از شاگرد ساده به  
درجه‌ی kyu(استادی) می‌رسی.



پلتفرم Codewars با گیمی‌فیکیشن تو رو معتاد به حل چالش  
و کلکل با باقی کد نویس‌ها می‌کنه.

## 4. Lmarena:

منبع بی‌نظیر از هوش  
مصنوعی‌های مطرح روز دنیا  
اونم به صورت رایگان و یکجا.



مثلاً توش می‌تونی Claude رو به  
جون Gemini بندازی و ببینی کی  
برنده رقابت میشه، رنکینگ و بنچمارک‌ها رو ببینی و مدل مناسب  
خودت رو پیدا کنی و همونجا مشغول استفاده ازش بشی.



### مهدی امامی

دانشجوی مهندسی کامپیوتر

دانشکده فابریک دانشگاه تهران

mahdiemami@ut.ac.ir



## نگاهی بر اخبار دنیای صفر و یک

۹۰ لا دقیقه

پس از جنجال‌های اوایل سال و معرفی مدل هوش مصنوعی DeepSeek، حالا مدل ریاضی این شرکت با نام Math-V2 معرفی شده که هدف آن حل مسائل پیچیده و گشودن رازهای جهان هستی عنوان شده است.

در DeepSeekMath-V2 تمرکز روی اثبات قضایا است. این مدل از یک چرخه تولید و اعتبارسنجی استفاده می‌کند؛ یعنی یک مدل اعتبارسنج LLM برای بررسی صحت اثبات‌ها آموزش دیده و سپس مولد اثبات نیز بر اساس همین اعتبارسنج تربیت شده است.

طبق گزارش‌ها، سیستم جدید قادر است به‌طور خودکار خطاها را شناسایی و اصلاح کند و حتی اثبات‌های سخت را برچسب‌گذاری می‌کند تا داده‌های تازه برای بهبود مداوم مدل جمع‌آوری شود.

موفقیت‌های Math-V2 هم چشمگیر بوده است:

🔴 کسب بالاترین امتیاز در IMO 2025 و CMO 2024

🔴 امتیاز شگفت‌انگیز ۱۱۸ از ۱۲۰ در آزمون Putnam 2024



سفر به اعماق اعداد با نهنگ چینی

رونمایی DeepSeek از Math-V2؛ هوش مصنوعی چینی‌ها برای

گشودن رمز و راز جهان هستی!

ماسک ادعا می‌کند تراشه‌های جدید می‌توانند توانایی‌های پزشکی و عملیاتی این ربات را به سطح خارق‌العاده‌ای برسانند؛ هرچند تاکنون نمایش عمومی قابل‌اعتمادی از عملکرد Optimus ارائه نشده است.



گیگابایت بر ثانیه کند شد!

سامسونگ سریع‌ترین حافظه HBM4 دنیا با پهنای باند 3.3 ترابایت بر ثانیه را ساخت.

سامسونگ بار دیگر معادلات بازار حافظه را تغییر داده و نسخه جدید HBM4 را آماده کرده است. حافظه‌ای که تا ۴۰ درصد سریع‌تر از نسل قبلی بوده و به پهنای باند خیره‌کننده ۳/۳ ترابایت بر ثانیه می‌رسد.

سامسونگ و SK Hynix در سال آینده رقابت فشرده‌ای برای عرضه نسل جدید حافظه‌های HBM خواهند داشت. بازاری که حالا بیش از هر زمان دیگری برای هوش مصنوعی، دیتاسنترها و شتاب‌دهنده‌های محاسباتی اهمیت حیاتی دارد.



وقتی پولات زیادی کرده

ادعای بزرگ ایلان ماسک: می‌خواهیم بیش از مجموع تمام تراشه‌سازان دنیا تراشه بسازیم!

ایلان ماسک اعلام کرده است که تسلا نسل جدید تراشه‌های اختصاصی هوش مصنوعی خود با نام AI5 را آماده کرده و کار روی AI6 نیز آغاز شده است. او می‌گوید هدف تسلا این است که «بیش از مجموع تمام شرکت‌های دیگر» تراشه تولید کند؛ ادعایی که با توجه به غول‌هایی مثل انویدیا، اینتل و AMD بسیار بلندپروازانه به نظر می‌رسد.

ماسک می‌گوید این تراشه‌ها توسط تیم مهندسی پیشرفته تسلا توسعه یافته‌اند و شرکت برای تولید انبوه آنها حتی احتمال ساخت کارخانه اختصاصی خود با نام TeraFab را بررسی می‌کند.

بخش دیگری از این برنامه مربوط به ربات انسان‌نمای Optimus است.



عجب موزی

**Nano Banana Pro** وارد شد؛ پیشرفته‌ترین مدل تولید عکس با هوش مصنوعی گوگل!

گوگل امروز اعلام کرد که مدل هوش مصنوعی تولید تصویر Nano Banana Pro را به مدل Nano Banana Pro ارتقا داده است تا قابلیت‌های بهتری را در دسترس قرار دهد. این مدل تولید عکس از هوش مصنوعی Gemini 3 بهره می‌برد.

مدل تولید تصویر با استفاده از هوش مصنوعی Nano Banana Pro که امروز از سوی گوگل معرفی شد دارای قابلیت‌های ویرایشی جدید، امکان ساخت تصاویر با وضوح بالاتر، رندر دقیق‌تر متن و امکان جستجو در وب است.

گوگل مدعی است که Nano Banana Pro نسبت به نسخه قبلی یعنی Nano Banana، تصاویر دقیق‌تر و متن‌های خواناتری را تولید می‌کند و قابلیت ایجاد نوشتار با سبک‌ها، فونت‌ها و زبان‌های متنوع را در اختیار دارد.



بعد از ۱۰ دیگه ۱۱ نیست

هجوم کاربران ویندوز ۱۰ به لینوکس ادامه دارد؛ Bazzite هم رکورد زد!

پایان پشتیبانی از ویندوز ۱۰ به یک فرصت طلایی برای لینوکس تبدیل شده است. بعد از آمار خیره‌کننده Zorin OS، حالا نوبت Bazzite است که خبرساز شود.

۲.۳ میلیون تلاش هک را مسدود کند و نرخ آلوده شدن بازی‌ها را به تنها ۲٪ کاهش دهد.

ترکیب هوشمند Javelin با قابلیت Secure Boot در مادربردها، شناسایی هکرها را بسیار دقیق و سریع کرده و تجربه‌ای امن برای بازیکنان ایجاد کرده است.

طبق گزارش رسمی EA، سیستم ضد تقلب سطح کرنل این شرکت، EA Javelin، با بررسی‌های پرتعداد در زمان آزمایش‌های بتای باز قبل از عرضه، امکان بهبود سیستم و افزایش بازی‌های بدون هکرا فراهم کرده و درصد بازی‌های بدون هکرا از ۹۳٪ به ۹۸٪ رسانده است.



ماشین بخار واقعی

ارزان‌تر از یک PC گیمینگ؛ قیمت احتمالی Steam Machine مشخص شد.

شرکت Valve اوایل ماه جاری Steam Machine را معرفی کرد و مدعی شد که این دستگاه در سال ۲۰۲۶ از راه می‌رسد؛ با این وجود، همچنان خبری از برچسب قیمت این دستگاه نیست.

در ویدیوی جدیدی از Linus Tech Tips، تیم لاینوس تلاش کرده با ساخت یک سیستم مشابه از نظر مشخصات، قیمت احتمالی Steam Machine را تخمین بزند. بر اساس محاسبات اعضای این کانال یوتیوب، اگر ارزان‌ترین قیمت‌های قطعات در نظر گرفته شود هزینه ساخت دستگاه برای Valve حدود ۶۰۲ دلار خواهد بود.



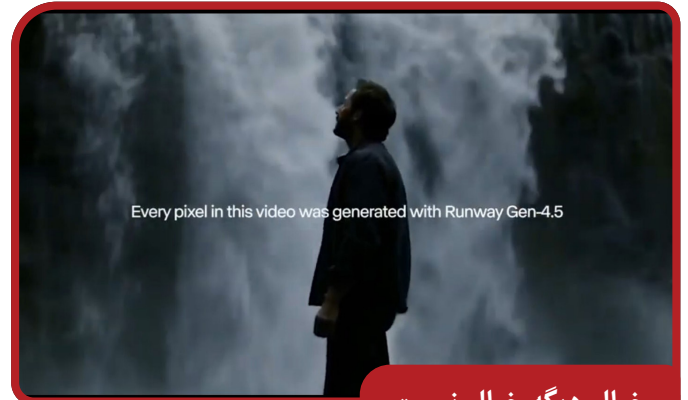
نامیرا تر از گرشاسپ

آبدیت جدید برای نمیرترین کارت گرافیک دنیا منتشر شد؛ RX 580 با بایوس جدید!

شرکت XFX پس از گذشت ۷ سال از عرضه کارت گرافیک افسانه‌ای Radeon RX 580، به‌طور غیرمنتظره یک BIOS جدید برای مدل 4GB

توسعه‌دهندگان این توزیع گیمینگ اعلام کردند تنها در یک ماه بیش از یک پتابایت دانلود ثبت شده است! با توجه به حجم میانگین ۷ گیگابایتی هر ISO، یعنی حدود ۱۴۳ هزار نصب جدید.

در طرف دیگر، Zorin OS هم با محیط شبیه ویندوز و پشتیبانی ساده از برنامه‌های قدیمی با WINE مهاجرت کاربران را آسان‌تر کرده. حالا هم آمارها می‌گویند نزدیک ۹۰٪ بازی‌های ویندوز روی لینوکس قابل اجرا هستند.



خیال دیگه خیال نیست

هوش مصنوعی ویدیوساز Runway Gen-4.5 معرفی شد؛ قدرتمندتر از Veo و Sora!

سرانجام Runway Gen-4.5، مدل هوش مصنوعی ویدیوساز جدید، معرفی شد. این مدل با درک دقیق قوانین فیزیک و جزئیات بصری خیره‌کننده، مرز بین واقعیت و ویدیوهای ساختگی را کمتر از همیشه کرده و در بنچمارک‌ها از مدل‌های گوگل و OpenAI پیشی گرفته است.

ویژگی‌های برجسته Gen-4.5:

- شبیه‌سازی دقیق وزن، تکانه و نیرو اشیا
- جریان طبیعی مایعات و رفتار واقعی اجسام
- اجرای پرامپت‌های متنی پیچیده بدون از دست دادن جزئیات



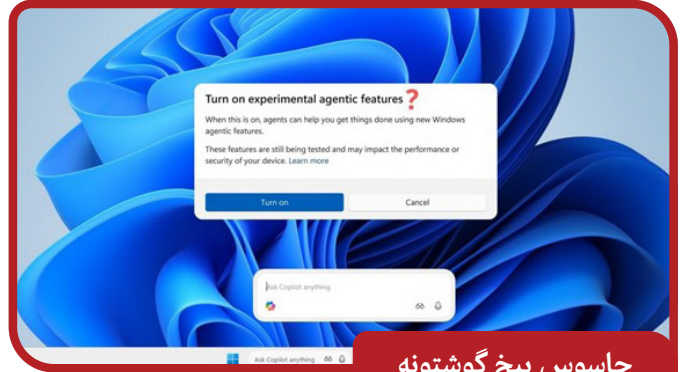
چیتربی چیترب

چرا Battlefield 6 به‌سادگی هک نمی‌شود؟

بتلفیلد ۶ با سیستم ضد تقلب سطح کرنل EA Javelin توانسته بیش از

# پیچ‌نوت

این کارت منتشر کرده است؛ حرکتی که کاربران را حسابی غافلگیر کرده است. این کارت قدیمی با معماری Polaris هنوز هم در سیستم‌های اقتصادی و دست‌دوم حضور پررنگی دارد و حتی در دوران ماینینگ یکی از محبوب‌ترین انتخاب‌ها بود. حالا با این آپدیت عجیب، دوباره نامش بر سر زبان‌ها افتاده است.



جاسوس بیخ گوش‌تونه

گل بود به سبزه نیز آراسته شد؛ ایجننت‌های AI ویندوز ۱۱ هم خطاکار و هم هذیان‌گو!

ویندوز ۱۱ وارد عصر جدیدی شده است؛ دورانی که مایکروسافت آن را «سیستم‌عامل عامل‌محور» می‌نامد. عامل‌های هوش مصنوعی با حساب‌های جداگانه در پس‌زمینه اجرا می‌شوند و از اجرای برنامه‌ها تا جست‌وجو، خرید و مدیریت فایل‌ها را انجام می‌دهند.

اما مشکل جدی اینجاست: مایکروسافت رسماً اعلام کرده این عامل‌ها ممکن است دچار خطای ادراکی شوند، خروجی‌های غیرمنتظره تولید کنند و حتی با حملاتی مانند Cross Prompt Injection فریب خورده و بدافزار نصب کنند یا فایل‌های حساس کاربر را ارسال کنند.

مایکروسافت برای کاهش ریسک، حساب‌های محدود، دسترسی کنترل‌شده و ثبت کامل فعالیت عامل‌ها را معرفی کرده است. با این حال، ویژگی‌های عامل‌محور هنوز «آزمایشی» هستند و خود شرکت تأکید می‌کند که فعال‌سازی آن‌ها با ریسک‌های جدی همراه است.



تیم قرمز و آبی ۸ تایی شدن

بازار گرافیک در سه‌ماهه سوم ۲۰۲۵؛ رشد جزئی AMD و اینتل در سایه سلطه‌ی انویدیا:

طبق جدیدترین گزارش موسسه JPR، بازار ۸.۸ میلیارد دلاری کارت

گرافیک مجزا در سه‌ماهه سوم سال ۲۰۲۵ شاهد تغییرات جالبی بوده است. با اینکه انویدیا همچنان حاکم بلامنازع بازار است، AMD و اینتل موفق شدند تکه‌های کوچکی از کیک بازار را از آن خود کنند.

تغییرات سهم بازار:

انویدیا: ۹۲٪ (کاهش ۱.۲ درصدی)

ای‌ام‌دی: ۷٪ (رشد ۰.۸ درصدی)

اینتل: ۱٪ (رشد ۰.۴ درصدی)

در این بازه زمانی، حدود ۱۲ میلیون کارت گرافیک عرضه شده است که رشدی ۲.۸ درصدی نسبت به فصل قبل را نشان می‌دهد. البته این رشد کمتر از میانگین ۱۰ ساله است؛ چرا که ترس از تعرفه‌ها در فصل قبل باعث خریدهای هیجانی و پیش‌خور شدن بازار شده بود.



خنجر از پشت؟

رقیب غیرمنتظره‌ی انویدیا؛ بزرگ‌ترین شریک غول سبز، تولید TPUهای گوگل را آغاز کرد.

بازی دوسربرد فاکس‌کان:

فاکس‌کان، بزرگ‌ترین شریک تجاری انویدیا، حالا سفارش‌های سنگینی برای تولید کلاسترهای هوش مصنوعی گوگل دریافت کرده است. طبق گزارش‌های تایوانی، فاکس‌کان مسئولیت تولید رک‌های پردازشی گوگل را برعهده گرفته است تا به بازیگری کلیدی در اکوسیستم سخت‌افزاری گوگل تبدیل شود.

هیولای نسل هفتم گوگل:

گوگل با نسل هفتم TPU و پلتفرم Ironwood، فقط سرعت را هدف نگرفته است، بلکه «مقیاس‌پذیری» را هم مدنظر داشته است. زیرساخت Superpod این شرکت گنجایش ۹۲۱۶ تراشه با توان پردازشی خیره‌کننده ۴۲.۵ اگرافلاپس (FP8) را دارد.



# کلام آخر

از همه شما مخاطبان عزیز نشریه دنیای صفر و یک، دعوت می‌کنیم که مطالب و مقالات خود را در حوزه‌های مربوط به مهندسی کامپیوتر شامل موضوعات تخصصی رشته و یا موضوعات بین‌رشته‌ای برای ما ارسال کنید. مقالات پس از داوری توسط داوران انتخاب شده و پس از بررسی نهایی به مرحله انتشار می‌رسند. مقالات خود را می‌توانید با دو شیوه زیر ارسال کنید:

۱. **وبسایت نشریه:** ابتدا در سایت نشریه ([cesasj.ut.ac.ir](http://cesasj.ut.ac.ir)) ثبت نام کنید. پس از تکمیل اطلاعات کاربری از قسمت ارسال مقالات، مقاله خود را ارسال کنید.

۲. **از طریق تلگرام:** فایل اصلی مقاله به همراه مشخصات نویسندگان را به آیدی ([@Cesa\\_Admin](https://t.me/Cesa_Admin)) ارسال کنید. گفتنی است که از سمت نشریه، گواهی پذیرش و انتشار مقالات به نویسندگان اعطا خواهد شد و مقالات منتشرشده از طریق وبسایت نشریه به صورت دوزبانه قابل رهگیری خواهند بود.

اگر علاقه‌مند به عضویت در کارگروه‌های مختلف نشریه اعم از: کارگروه تخصصی، کارگروه اخبار، کارگروه فیلم و سریال و ... یا بخش‌های اجرایی نشریه مانند ویراستاری، صفحه‌آرایی و ... هستید، نیز می‌توانید از طریق راه‌های ارتباطی زیر با ما در ارتباط باشید.

امیدواریم از مطالعه این شماره از نشریه دنیای صفر و یک لذت برده باشید. پذیرای نظرات، انتقادات و پیشنهادات شما هستیم.

ایمیل: [donyayesefreyek.ut@gmail.com](mailto:donyayesefreyek.ut@gmail.com)

وبسایت: [cesasj.ut.ac.ir](http://cesasj.ut.ac.ir)

ایتا: [@donyayesefreyek](https://t.me/donyayesefreyek)

تلگرام: [@donyayesefreyek](https://t.me/donyayesefreyek)

لینکدین: [linkedin.com/company/donyayesefreyek](https://www.linkedin.com/company/donyayesefreyek)

برای اطلاع از اخبار و فعالیت‌های انجمن و نشریه، صفحات مجازی انجمن را دنبال کنید.

ایتا: [@CESA\\_UT](https://t.me/CESA_UT)

تلگرام: [@CESA\\_UT](https://t.me/CESA_UT)

باعث خرسندی ماست که این شماره از نشریه را با چشمانتان بدرقه کردید. به امید پویایی و حس‌وحال مثبت برای جامعه تحصیلی و دانشگاه‌های کشور.

برای قدم زدن در دنیای صفر و یک با ما همراه باشید.

# شماره‌های قبلی دنیای صفر و یک:



صفحات مجازی ما را دنبال کنید.

@CESA\_UT

CESA\_UT

