



واحد اردبیل

شماره ۱ | تابستان ۱۴۰۵

فصلنامه تخصصی

شوتک

رسانه فناوری های نوین



نئوتک

NEOTECH

نشریه تخصصی و فصلنامه دانشجویی
دانشگاه آزاد اسلامی واحد اردبیل



شماره :
اول



تاریخ انتشار :
تابستان ۱۴۰۵



دوره :
اول



تناوب انتشار :
فصلنامه

تاریخ مجوز نشریه : ۱۴۰۵/۰۵/۰۸

شماره مجوز نشریه : ۱۹۶۱/۴۰۵۹/د

زمینه فعالیت :
کامپیوتر و فناوری های نوین



صاحب امتیاز
انجمن علمی مهندسی کامپیوتر



مدیر مسئول
مهدی بخشی



سردبیر نشریه
محمدرضا رجبی ترکمانی



گروه تحریر

علی محمد سرخانیان
محمد مهدی عباس زاده
سحر سیدایمانی

کیمیا رهنما آذر | دبیر گروه تحریر
مهران جاوید



گروه ویراستاری

محمدرضا رجبی | دبیر ویراستاری
علی باختر
بردیا دورنما
هلیا جهدی



گروه طراحی و گرافیک

مهدی بخشی | دبیر گروه طراحی
آراد لطف الهیان
سانای جاهدی
کیمیا رهنما آذر



realneotech



realneotech



realneotech.ir



انجمن علمی مهندسی کامپیوتر
دانشگاه آزاد اسلامی
واحد اردبیل



فهرست مطالب

	معرفی اعضا	صفحه	۴
	مقدمه	صفحه	۶
	اخبار تکنولوژی	صفحه	۹
	سه ضلعی طلایی تحلیل داده در پایتون	صفحه	۱۲
	مصاحبه	صفحه	۲۱
	بهترین ابزار های هوش مصنوعی به تفکیک نیاز	صفحه	۲۳
	perlin noise	صفحه	۲۶
	پرسود ترین زیر شاخه های علوم کامپیوتر	صفحه	۳۴
	تگ های پرکاربرد HTML , CSS	صفحه	۴۰
	boot .dev	صفحه	۴۹
	وقت اضافه	صفحه	۵۳
	دورنگاه	صفحه	۵۵
	مقاله علمی	صفحه	۶۲
	Unreal Engine 6	صفحه	۶۶
	پادکست	صفحه	۶۹
	کامیک کان	صفحه	۷۰
	سخن بزرگان	صفحه	۷۳

نئوتک؛ فصلنامه‌ای برای نگاهی به آینده را نمی‌بینید، بلکه آن را می‌سازیم.



معرفی اعضا

ترکمانی

محمدرضا رجبی

- سردبیر نشریه
- دبیر گروه ویراستاری
- مسئول همایش های انجمن علمی مهندسی کامپیوتر
- ترم ۵ مهندسی کامپیوتر



مهدی بخشی

- مدیر مسئول نشریه
- دبیر گروه طراحی و گرافیک
- روابط عمومی انجمن علمی مهندسی کامپیوتر
- ترم ۵ مهندسی کامپیوتر



مهران جاوید

- عضو گروه تحریریه
- تحریر بخش های مربوط به بازی ، ۳ کتابخانه برتر پایتون ، Perlin noise و Boot.Dev
- ترم ۵ مهندسی کامپیوتر



کیمیا رهنماآذر

- دبیر گروه تحریریه
- عضو گروه طراحی و گرافیک
- دبیر انجمن علمی مهندسی کامپیوتر
- ترم ۵ مهندسی کامپیوتر



هلیا جهدی

- عضو گروه ویراستاری
- ویراستاری HTML و CSS
- نائب دبیر انجمن علمی مهندسی کامپیوتر
- ترم ۵ مهندسی کامپیوتر



آراد لطف الهیان

- عضو گروه طراحی و گرافیک
- طراحی بخش های مربوط به ۳ کتابخانه برتر پایتون ، هوش مصنوعی ، CSS و HTML
- ترم ۵ مهندسی کامپیوتر



معرفی اعضا

سانای جاهدی

- عضو گروه طراحی و گرافیک
- طراحی بخش فهرست مطالب
- بازرس انجمن علمی
- مهندسی کامپیوتر
- ترم ۵ مهندسی کامپیوتر



محمد مهدی عباس زاده

- عضو گروه تحریریه
- تحریر بخش های مربوط به مقاله علمی DEVOPS و پرسودترین رشته کامپیوتر
- ترم ۵ مهندسی کامپیوتر



علی محمد سرخانیان

- عضو گروه تحریریه
- طراح سایت نئوتک
- تحریر بخش های CSS و HTML
- کارشناسی رشته حقوق
- ترم ۵ مهندسی کامپیوتر



علی باختر

- عضو گروه ویراستاری
- ویراستاری بخش های مربوط به Perl Noise و Boot.Dev
- ترم ۵ مهندسی کامپیوتر



سحر سید ایمانی

- عضو گروه تحریریه
- تحریر بخش هوش مصنوعی
- مسئول کارگاه های انجمن علمی مهندسی کامپیوتر
- ترم ۵ مهندسی کامپیوتر



بردیا دورنما

- عضو گروه ویراستاری
- ویراستاری بخش های مربوط به دور نگاه ، کوانتوم و Unreal Engine 6
- ترم ۵ مهندسی کامپیوتر



مقدمه مدیر مسئول

گاهی ایده ای که در ذهن شماست ، یک ایده نمی ماند. تبدیل می شود به چیزی که برای ساختنش وقت می گذارید ، یاد می گیرید، تجربه می کنید و قدم به قدم به آن شکل می دهید. برای من ، نئوتک از همین جا شروع شد ؛ از یک ایده و دغدغه برای ساختن رسانه ای دانشجویی در دنیای کامپیوتر و فناوری . ایده ای که کم کم تبدیل شد به مسیری برای کار تیمی و ساختن چیزی که به آن افتخار کنیم. نئوتک قرار است فضایی باشد که در آن فناوری ، فقط موضوعی برای خواندن نباشد بلکه فرصتی باشد برای یاد گرفتن و تجربه کردن. ما می خواهیم از نگاه یک دانشجو به دنیای فناوری نگاه کنیم ؛ کنجکاوانه ، یادگیرنده و رو به آینده . از اتفاقاتی که شاید امروزه در حال شکل گرفتن باشند ولی فردای ما را می سازند .

نئوتک ، نشریه دانشگاه آزاد اسلامی واحد اردبیل با صاحب امتیازی انجمن علمی مهندسی کامپیوتر ، برای روایت دنیای کامپیوتر و تکنولوژی. این شماره ، اولین گام ما در مسیری است که هنوز ابتدای آن ایستاده ایم ؛ مسیری که امیدواریم با هر شماره تجربه های بیشتری به دست بیاوریم ، چیزهای بیشتری یاد بگیریم و نئوتک را بهتر و پخته تر از قبل بازسازی کنیم. در پایان ، از جناب آقای دکتر حسن حسین زاده ، جناب آقای دکتر بابک نوری مقدم و جناب آقای بهنام شوقی بابت همراهی و کمک شایان به این نشریه و دانشجویان جناب آقای مهدی نامور سوها ، جناب آقای آیدین کاوه و سرکار خانم آيسان فرج الهی بابت کمک در تألیف این نشریه و از تمام افرادی که در شکل گیری نخستین شماره نئوتک همراه ما بودند ، صمیمانه تشکر می کنیم؛ همچنین از اساتید و مسئولان دانشگاه که از این مسیر حمایت کردند تا تک تک اعضای تیم نئوتک که با وقت، ایده و انرژی خود این شماره را شکل دادند.

مهدی بخشی

مدیر مسئول نشریه نئوتک

مقدمه سردیر

فناوری با چنان شتابی پیش می‌رود که امروز حتی «دنبال کردن» آن نیز به مهارتی مستقل تبدیل شده است. هر روز ابزاری تازه معرفی می‌شود و مدلی جدید ظهور می‌کند؛ اما در میان این حجم از اطلاعات، مسئله فقط دانستن نیست، مهم‌تر از آن، درست دانستن است.

نئوتک با همین نگاه شکل گرفته است تا محتوایی دقیق، معتبر و ارزشمند را در اختیار مخاطب قرار دهد. در تهیه این شماره، هر مطلب از چند فیلتر اساسی گذشته است: اعتبار علمی، انسجام و ارزش واقعی برای مخاطب. برای ما «قابل قبول بودن» نقطه پایان نیست؛ معیار، قابل اعتماد بودن است و این را نه یک شعار، بلکه مسئولیتی حرفه‌ای می‌دانیم.

با این حال، نئوتک صرفاً مجموعه‌ای از مطالب نیست. این نشریه زمانی معنا پیدا می‌کند که مخاطب در آن حضور داشته باشد؛ بخواند، بپرسد، نقد کند و حتی با ما مخالف باشد. هر بازخورد، چه تحسین و چه نقد، قدمی است برای نزدیک‌تر شدن به آنچه یک نشریه تخصصی باید باشد.

از نویسندگان، ویراستاران، طراحان، اعضای تحریریه و دوستانی که با دقت و تعهد در شکل‌گیری این شماره نقش داشتند، صمیمانه سپاسگزارم. آنچه پیش روی شماست، حاصل کاری جمعی است که کیفیت را بر سرعت و عمق را بر سطحی‌نگری ترجیح داده است.

امیدوارم این شماره نه فقط اطلاعاتی تازه، بلکه نگاهی تازه به دنیای علم و فناوری به شما ارائه دهد؛ نگاهی که کنجکاو را بیشتر کند و انگیزه‌ای برای یادگیری و تجربه باشد.

محمد رضا رجبی ترکمانی

سردبیر نشریه نئوتک

مقدمه دبیر انجمن

دنیای ما، دنیای تغییرات شتابان است؛ جایی که مرز بین واقعیت و خیال در کدها و الگوریتم‌ها رنگ می‌بازد. ما در انجمن علمی مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد اردبیل، همواره بر این باور بوده‌ایم که رسالت دانشجو تنها محدود به کلاس‌های درس نیست؛ بلکه سهم‌خواهی از آینده، نیازمند بستری است برای خلق، نقد و اشتراک‌گذاری دانش.

نشریه «نئوتک»، حاصل ماه‌ها تلاش جمعیِ دوستان در انجمن علمی است. این نشریه، نه فقط یک رسانه، که یک «اکوسیستم علمی» است تا صدای دانشجویانی باشد که می‌خواهند فراتر از سرفصل‌های درسی، افق‌های جدیدی را در دنیای مهندسی ترسیم کنند. «نئوتک» آمده است تا پل ارتباطی باشد میان خلاقیت‌های دانشجویی و نیازهای روز دنیای تکنولوژی.

این شماره، آغاز یک مسیر است. مسیری که در آن قصد داریم پرسشگری، تفکر نقادانه و نوآوری را ترویج دهیم. از تمامی شما عزیزان دعوت می‌کنم تا «نئوتک» را متعلق به خود بدانید؛ قلم بزنید، نقد کنید و در ساختن این هویت جدید با ما همراه باشید.

در پایان، مراتب سپاس و قدردانی خود را تقدیم اعضای متعهد انجمن علمی مهندسی کامپیوتر، تیم خلاق تحریریه، تیم ویراستاری و طراحان هنرمندان می‌کنم که با اشتیاق خالصانه، این افکار را به واقعیت بدل کردند. همچنین از اساتید ارجمند جناب آقای دکتر حسن حسین زاده، خانم دکتر ندا قبادی، آقای بهنام شوقی و سایر عزیزان که با حمایت‌های معنوی، چراغ راه ما بودند، صمیمانه قدردانی می‌کنم.

کیمیا رهنماآذر

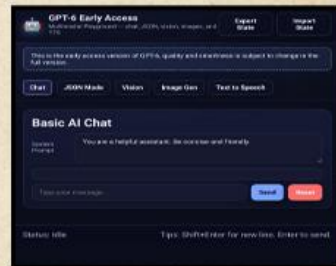
دبیر انجمن علمی مهندسی کامپیوتر

NEOTECH NEWS



اپل دوباره تاج ارزشمندترین شرکت جهان را گرفت

در تابستان ۲۰۲۶، رقابت میان دو غول فناوری یعنی Apple و NVIDIA برای عنوان ارزشمندترین شرکت جهان دوباره داغ شد. اپل در اواخر ژوئیه توانست ارزش بازار خود را به حدود ۴.۹ تریلیون دلار برساند و از NVIDIA عبور کند. این اتفاق پس از بیش از یک سال، جایگاه نخست را دوباره به اپل بازگرداند. این تغییر در شرایطی رخ داد که بازار نسبت به هزینه‌های سنگین زیرساخت هوش مصنوعی و بازده سرمایه‌گذاری شرکت‌های فعال در این حوزه حساس تر شده بود. در مقابل، اپل همچنان بر ترکیب سخت‌افزار، خدمات، اکوسیستم اختصاصی و هوش مصنوعی تکیه دارد.



6 Astra OpenAI GPT معرفی شد؛ وقتی AI خودش سراغ کارهای کامپیوتری می‌رود

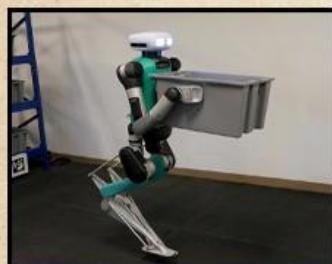
در سپتامبر ۲۰۲۶ از Astra GPT ۶ رونمایی کرد؛ مدلی که تمرکز آن فقط بر پاسخ‌دادن به سؤال‌ها نیست و توانایی‌های گسترده‌تری در کار با کامپیوتر، وب گردی، برنامه‌نویسی، امنیت سایبری و انجام کارهای حرفه‌ای دارد. OpenAI اعلام کرده Astra در آزمون‌های مختلف مربوط به استفاده از کامپیوتر و انجام وظایف پیچیده عملکرد بسیار بالایی داشته است. یکی از نکات مهم این نسل، حرکت از «چت‌بات» به سمت AI Agent است؛ یعنی مدلی که می‌تواند یک هدف را دریافت کند، مراحل انجام آن را مشخص کند و با ابزارهای مختلف برای رسیدن به نتیجه تعامل داشته باشد. اگر این روند ادامه پیدا کند، نقش کاربر از «انجام‌دهنده تمام مراحل» به ناظر و هدایت‌کننده فرایند تغییر خواهد کرد.

NEOTECH NEWS



AMD Helios از راه رسید؛ یک نیروگاه پردازشی برای عصر AI

AMD در رویداد Advancing AI ۲۰۲۶ مجموعه‌ای از محصولات جدید خود را معرفی کرد که مهم‌ترین آن‌ها سیستم رک‌محور AMD Helios بود. این شرکت همچنین از نسل ششم پردازنده‌های سروری EPYC، خانواده پردازنده‌های گرافیکی Instinct MI۴۰۰ و محصولات جدید Ryzen AI رونمایی کرد. Helios به جای اینکه فقط یک GPU یا CPU باشد، یک زیرساخت کامل پردازش هوش مصنوعی در مقیاس دیتاسنتر است. AMD اعلام کرده این سیستم برای اجرای مدل‌های بزرگ و بارهای کاری سنگین AI طراحی شده و شرکت‌هایی مانند OpenAI و Meta در حال آماده‌سازی برای استفاده از این زیرساخت هستند.

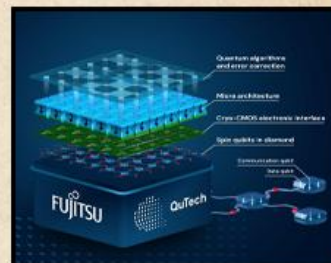


Digit5 معرفی شد؛ ربات انسان‌نمایی که برای محیط کار طراحی شده است

شرکت Agility Robotics در سپتامبر ۲۰۲۶ از نسل جدید ربات انسان‌نمای خود با نام Digit5 رونمایی کرد. این ربات برای فعالیت در محیط‌هایی مانند کارخانه‌ها، انبارها و مراکز لجستیک طراحی شده و هدف اصلی آن انجام کار در کنار انسان‌هاست.

Digit5 حاصل بیش از ۶۵ هزار ساعت فعالیت نسل‌های قبلی Digit در محیط‌های واقعی است. نسل جدید به گونه‌ای طراحی شده که بتواند بدون نیاز به موانع ایمنی سنتی، در نزدیکی کارکنان فعالیت کند و بخشی از فرایندهای کاری یک مجموعه را انجام دهد.

این اتفاق بخشی از روند بزرگ‌تری است که از ترکیب Robotics + AI + Computer Vision شکل گرفته است؛ جایی که ربات فقط یک دستگاه مکانیکی نیست و باید محیط اطراف خود را درک کرده و نسبت به آن واکنش نشان دهد.



نویسنده : مهدی بخشی

سه ضلعی طلایی تحلیل داده در پایتون

NumPy

NumPy که مخفف Numerical Python است، یکی از کتابخانه‌های اصلی پایتون در حوزه‌ی محاسبات علمی و فنی به‌شمار می‌رود. قابلیت‌های این کتابخانه در زمینه‌ی محاسبات آرایه‌محور، آن را به ابزاری کلیدی در حوزه‌های جبر خطی، تحلیل آماری و یادگیری ماشین تبدیل کرده است.

ویژگی‌های اصلی:

- ساختار داده‌ی `numpy.ndarray`، یک آرایه‌ی چندبعدی که امکان ذخیره‌سازی و پردازش داده‌های عددی را با کارایی بالا فراهم می‌آورد.
- مجموعه‌ای گسترده از توابع برای انجام عملیات عنصر به عنصر روی آرایه‌ها.
- پشتیبانی از عملیات جبر خطی نظیر ضرب ماتریس، تجزیه‌ی مقدار ویژه و حل معادلات خطی.

کاربردها در دنیای واقعی:

- تحلیل مالی (بهینه‌سازی سبد سرمایه‌گذاری)
 - پردازش تصویر (کار با آرایه‌های پیکسلی)
 - شبیه‌سازی‌های علمی (فیزیک، شیمی)
- یک آرایه‌ی NumPy یا `ndarray` مجموعه‌ای از عناصر هم نوع است که در ساختاری چندبعدی ذخیره می‌شوند. آرایه‌ها روشی کارآمد برای ذخیره‌سازی و انجام عملیات روی داده‌های عددی فراهم می‌کنند.
- تعداد ابعاد یک آرایه، «رتبه» (`rank`) نامیده می‌شود. اندازه‌ی آرایه در هر بُعد، «شکل» (`shape`) آن را مشخص می‌کند. دسترسی به عناصر با استفاده از کروشه‌های مربعی `[]` انجام می‌شود و آرایه‌ها معمولاً از لیست‌های پایتون ساخته می‌شوند.

مبانی نام‌پای

مدیریت آرایه‌ها

ایجاد و تغییر شکل آرایه‌ها

`np.array()`
`reshape()`

توابع ریاضی

انجام عملیات بر روی عناصر

`np.add()`
`np.multiply()`

جبر خطی

انجام عملیات ماتریسی

`np.dot()`
`np.linalg.inv()`

تولید اعداد تصادفی

تولید داده‌های تصادفی

`np.random.rand()`
`np.random.randint()`



ساخت آرایه در NumPy

آرایه‌های NumPy با استفاده از تابع `np.array()` ساخته می‌شوند. این آرایه‌ها را می‌توان از دنباله‌های پایتونی مانند لیست‌ها و تاپل‌ها ایجاد کرد.

```
import numpy as np

arr = np.array([1, 2, 3])
print(arr)

arr = np.array([[1, 2, 3], [4, 5, 6]])
print(arr)

arr = np.array((1, 3, 2))
print(arr)
```

توضیح:

- `np.array([1, 2, 3])` یک آرایه‌ی یک‌بعدی از یک لیست می‌سازد.
- `np.array([[1, 2, 3], [4, 5, 6]])` یک آرایه‌ی دوبعدی از یک لیست تو در تو ایجاد می‌کند.
- `np.array((1, 3, 2))` یک آرایه‌ی یک‌بعدی از یک تاپل می‌سازد.

دسترسی به اندیس آرایه

عناصر یک آرایه‌ی NumPy را می‌توان با استفاده از اندیس‌گذاری (indexing) و برش زنی (slicing) دسترسی پیدا کرد. اندیس‌گذاری عناصر مشخص را بازیابی می‌کند، در حالی که برش‌زنی محدوده‌ای از عناصر را از آرایه استخراج می‌کند.

```
import numpy as np

arr = np.array([
    [-1, 2, 0, 4],
    [4, -0.5, 6, 0],
    [2.6, 0, 7, 8],
    [3, -7, 4, 2.0]
])

arr2 = arr[:2, ::2]
print("دو سطر اول و ستون‌های یک‌درمیان:\n", arr2)

arr3 = arr[[1, 1, 0, 3], [3, 2, 1, 0]]
print("عناصر انتخاب‌شده:", arr3)
```

توضیح:

- `arr[:2, ::2]` دو سطر اول و ستون‌های یک‌درمیان (ستون‌های ۰ و ۲) را انتخاب می‌کند.
- `arr[[1, 1, 0, 3], [3, 2, 1, 0]]` عناصر مشخص‌شده توسط جفت‌های اندیس (سطر، ستون) را بازیابی می‌کند.



عملیات پایه روی آرایه‌ها

آرایه‌های NumPy از عملیات ریاضی متنوعی پشتیبانی می‌کنند که می‌توانند روی عناصر تکی یا بین چندین آرایه اعمال شوند. این عملیات بدون استفاده از حلقه‌های صریح، با کارایی بالا اجرا می‌شوند.

```
import numpy as np

a = np.array([[1, 2],
              [3, 4]])

b = np.array([[4, 3],
              [2, 1]])

print("افزودن ۱ به هر عنصر:\n", a + 1)
print("کم کردن ۲ از هر عنصر:\n", b - 2)
print("جمع تمام عناصر آرایه:", a.sum())
print("جمع آرایه‌ها:\n", a + b)
```

توضیح:

- $a + 1$ به هر عنصر آرایه‌ی a یک واحد اضافه می‌کند.
- $b - 2$ از هر عنصر آرایه‌ی b دو واحد کم می‌کند.
- `a.sum()` مجموع تمام عناصر موجود در a را برمی‌گرداند.
- $a + b$ جمع عنصر به عنصر (element-wise) دو آرایه را انجام می‌دهد.

نوع داده در NumPy:

هر آرایه‌ی NumPy دارای یک نوع داده (dtype) است که نوع مقادیر ذخیره‌شده در آرایه را تعیین می‌کند NumPy. می‌تواند به‌طور خودکار نوع داده را تشخیص دهد، یا می‌توانید هنگام ساخت آرایه، آن را به‌صراحت مشخص کنید.

```
import numpy as np

x = np.array([1, 2])
print(x.dtype)

x = np.array([1.0, 2.0])
print(x.dtype)

x = np.array([1, 2], dtype = np.int64)
print(x.dtype)
```

توضیح:

- `np.array([1, 2])` یک آرایه‌ی صحیح می‌سازد، بنابراین نوع داده‌ی آن `int64` است.
- `np.array([1.0, 2.0])` یک آرایه‌ی اعشاری می‌سازد، بنابراین نوع داده‌ی آن `float64` است.
- `dtype=np.int64` به‌صراحت نوع داده‌ی آرایه را به `int64` تنظیم می‌کند.

- [یک آرایه‌ی اعشاری می‌سازد، بنابراین نوع داده‌ی آن float64 است.
- dtype=np.int64 به‌صراحت نوع داده‌ی آرایه را به int64 تنظیم می‌کند.
- ویژگی dtype برای مشاهده‌ی نوع داده‌ی یک آرایه‌ی NumPy استفاده می‌شود.

عملیات ریاضی روی آرایه‌های با نوع داده‌ی مشخص

NumPy توابع توکاری برای انجام عملیات ریاضی روی آرایه‌ها ارائه می‌دهد. این عملیات به‌صورت عنصر به عنصر اعمال می‌شوند و می‌توانند به‌طور هم‌زمان روی کل آرایه اجرا شوند.

توضیح:

- np.add(arr1, arr2) جمع عنصر به عنصر دو آرایه را انجام می‌دهد.
- np.sum(arr1) مجموع تمام عناصر موجود در arr1 را برمی‌گرداند.
- np.sqrt(arr1) جذر هر عنصر از arr1 را محاسبه می‌کند.
- arr1.T ترانهاده‌ی آرایه را با جابجایی سطرها و ستون‌ها برمی‌گرداند.

Pandas

پانداس (که مخفف Python Data Analysis است) یک کتابخانه‌ی نرم‌افزاری متن‌باز (open-source) است که برای دستکاری و تحلیل داده طراحی شده است. این کتابخانه بر روی NumPy ساخته شده و به‌طور کارآمدی داده‌های حجیم را مدیریت می‌کند و ابزارهایی برای پاک‌سازی، تبدیل و تحلیل داده در اختیار کاربران قرار می‌دهد. پانداس به‌خوبی با سایر کتابخانه‌های پایتون مانند NumPy، Matplotlib و scikit-learn یکپارچه می‌شود و حول دو ساختار داده‌ی اصلی شکل گرفته است: Series (یک‌بعدی) و DataFrame (دو‌بعدی). همچنین روش‌هایی مانند dropna و fillna را برای مدیریت روانِ مقادیر گم‌شده ارائه می‌دهد.

ویژگی‌های کلیدی:

- پانداس دارای ساختارهای داده‌ی DataFrame و Series است که برای کار با داده‌های جدولی دو‌بعدی و آرایه‌های یک‌بعدی طراحی شده‌اند.
- پانداس ابزارهای خاصی برای کار با داده‌های سری‌های زمانی (time series) ارائه می‌دهد.
- پانداس دارای ابزارهایی برای مدیریت داده‌های گم‌شده (missing data)، شناسایی و حذف داده‌های تکراری (duplication) و سایر عملیات پاک‌سازی داده است.



کاربردهای عملی:

- **پاک‌سازی و گزارش‌دهی داده‌های کسب‌وکار**
- **تحلیل سری‌های زمانی در بازار سهام**
- **دستکاری داده‌های CSV و Excel در استارت‌آپ‌ها**

نکات مهم:

DataFrame: یک ساختار داده‌ی دوبعدی با سطرها و ستون‌هاست که شباهت زیادی به صفحه‌گسترده‌ی Excel دارد.

ریشه‌ی نام: نام Pandas برگرفته از اصطلاح «panel data» در اقتصادسنجی است که به مجموعه داده‌های چندبعدی اشاره دارد.

یانداس برای چه کاری استفاده می‌شود؟

- خواندن و نوشتن داده از فرمت‌های مختلف مانند CSV ، Excel و پایگاه‌های داده‌ی SQL.
- پاک سازی و آماده‌سازی داده (مدیریت مقادیر گم‌شده، فیلتر کردن، حذف داده‌های تکراری).
- ادغام (merging) ، اتصال (joining) و تغییر شکل (reshaping) مجموعه داده‌ها.
- انجام تحلیل‌های آماری و محاسبه‌ی آمارهای توصیفی.
- مصورسازی سریع داده.

DataFrame

یک DataFrame یک ساختار داده‌ی دوبعدی، تغییرپذیر از نظر اندازه و بالقوه ناهمگن است که دارای محورهای برچسب‌دار (سطرها و ستون‌ها) می‌باشد.

- ایجاد DataFrame
- ایندکس در DataFrame
- دسترسی به DataFrame
- ایندکس گذاری و انتخاب داده در پانداس
- برش زنی (Slicing) در DataFrame
- فیلتر کردن DataFrame با چند شرط
- ادغام، اتصال و الحاق (Concatenating) DataFrames
- مرتب سازی DataFrame

- ادغام، اتصال و الحاق (Concatenating) DataFrames

- مرتب‌سازی DataFrame

- جدول محوری (Pivot Table) در پانداس

یک Series یک آرایه‌ی برچسب‌دار یک‌بعدی است که می‌تواند هر نوع داده‌ای (اعداد صحیح، رشته‌ها، اعداد اعشاری، اشیای پایتون و غیره) را در خود نگه دارد. این ساختار مشابه یک ستون در صفحه‌گسترده یا جدول پایگاه داده است.

- ایجاد Series

- دسترسی به عناصر یک Series

- عملیات دودویی (Binary Operations) روی Series

- متدهای Index() در Series

پاک‌سازی داده‌ها

حذف داده‌های تکراری و اصلاح ناهماهنگی ستون‌ها.

قبل

نام	سن	میزان_حقوق
Aryan	25	5000
Harsh	30	7000
Aryan	25	5000

بعد

نام	سن	حقوق
Aryan	25	5000
Harsh	30	7000

حذف ردیف‌های تکراری با (drop_duplicates)
و تغییر نام ستون‌ها با (rename)

عملیات اعمال‌شده:

پانداس توابع متنوعی برای خواندن و نوشتن داده از فرمت‌های مختلف ارائه می‌دهد:

- خواندن فایل‌های CSV با پانداس

- نوشتن داده در فایل‌های CSV

- خروجی DataFrame به فایل CSV

- خواندن فایل‌های JSON با پانداس

- تجزیه (Parsing) مجموعه داده‌های JSON

- خروجی DataFrame به فایل JSON

- کار با فایل‌های Excel در پانداس

- خواندن فایل‌های متنی با پانداس

File

Edit

Selection

View

Go

Run

Terminal

Help

Neotek

×

+

...

پاک‌سازی داده، گامی ضروری در پیش‌پردازش (preprocessing) داده برای تضمین دقت و یکنواختی است:

- مدیریت داده‌های گم‌شده
- حذف داده‌های تکراری
- تغییر نوع داده (datatype) در پانداس
- حذف ستون‌های خالی
- دستکاری رشته‌ها در پانداس
- متدهای رشته‌ای در پانداس
- تشخیص انواع داده‌ی مختلط و رفع آن

عملیات پیشرفته

- یافتن همبستگی (Correlation) بین داده‌ها
- مصورسازی داده با پانداس
- توابع رسم (Plotting) پانداس برای مصورسازی داده
- مبانی کار با سری‌های زمانی با پانداس
- تحلیل و مصورسازی سری‌های زمانی در پایتون

پروژه‌ها

پروژه‌های تحلیل داده‌ی واقعی با استفاده از پانداس و سایر ابزارهای علوم داده:

- تحلیل داده‌های Zomato با پایتون
- تحلیل داده‌های IPL
- تحلیل داده‌های Airbnb
- تحلیل و مصورسازی داده‌های جهانی کووید-۱۹
- تحلیل و پیش‌بینی قیمت مسکن
- تحلیل سبد خرید (Market Basket Analysis)
- تحلیل مجموعه‌داده‌ی تایتانیك و پیش‌بینی بقا
- تحلیل و پیش‌بینی مجموعه‌داده‌ی گل زنبق (Iris)
- تحلیل ریزش مشتری (Customer Churn)

main

⊗ 0

⚠ 0

Ln 1, Col 1

Spaces: 4

UTF-8

LF

FileEditSelectionViewGoRunTerminalHelp

Neotek

Matplotlib

مَت پلات لیب (Matplotlib) یک کتابخانه‌ی مصورسازی داده (data visualization) است که به توسعه‌دهندگان امکان می‌دهد تا تصاویر ایستا (static)، پویا (animated) و تعاملی (interactive) را در پایتون ایجاد کنند. نمودارها و پلات‌های تولیدشده توسط این کتابخانه، به‌طور گسترده برای مصورسازی داده مورد استفاده قرار می‌گیرند.

ویژگی‌های کلیدی:

پشتیبانی از پلات‌های خطی، نمودارهای میله‌ای، هیستوگرام‌ها، نمودارهای پراکندگی و مصورسازی‌های سه‌بعدی. کنترل کامل بر عناصر شکل (figure) شامل محورها (axes)، برچسب‌ها (labels)، راهنما (legends) و رنگ‌ها. طراحی شده برای کار با آرایه‌های NumPy و DataFrame‌های پانداس. سازگار با چندین ابزارک عمومی رابط کاربری گرافیکی (wxPython و GTK، Qt، Tkinter). قابلیت تولید خروجی با کیفیت بالا در قالب‌های چاپی مختلف (SVG، PDF، PNG). کاربردهای عملی:

- مقالات پژوهشی دانشگاهی
- نمودارهای ایستای داشبوردها
- تحلیل اکتشافی داده (EDA) در دفترچه‌های نوت‌بوک

چرا از Matplotlib استفاده کنیم؟

- **ماژول : Pyplot** یک رابط رویه‌ای (procedural) شبیه به MATLAB ارائه می‌دهد که تولید نمودارها را برای اکتشاف سریع داده ساده می‌کند.
- **معماری (Figure) در برابر Axes** یک **Figure** به‌عنوان ظرف کلی سطح‌بالا عمل می‌کند، در حالی که **Axes** به منطقه‌ی واقعی که داده در آن رسم می‌شود اشاره دارد – این امکان را می‌دهد که چندین نمودار در یک پنجره قرار گیرند.
- **سفارشی‌سازی سطح‌بالا**: کنترل دقیق بر زیبایی‌شناسی بصری از جمله عنوان‌ها، راهنماها، نقشه‌های رنگی (maps و چیدمان‌های پیچیده‌ی شکل) زیرنمودارها یا subplots را فراهم می‌کند.
- **ابزارک‌های تعاملی**: شامل ماژول اختصاصی برای افزودن لغزنده‌ها (sliders)، دکمه‌ها و چک‌باکس‌ها به‌طور مستقیم روی نمودارها برای دستکاری پویای داده است.

main 0 0Ln 1, Col 1Spaces: 4UTF-8LF

سفارشی‌سازی نمودارها

به ما امکان می‌دهد تا ویژگی‌های Figure و Axes را دستکاری کنیم. تنظیم چیدمان فضایی، تایپوگرافی، مشخصات شبکه (grid) و سطوح شفافیت (alpha)

- نشانگرها (Markers)
- افزودن برچسب‌ها
- پیکربندی شبکه
- ایجاد زیرنمودارها (Subplots)
- استایل‌دهی به نمودارها
- تغییر اندازه‌ی نمودار
- تنظیم شفافیت نمودار
- تغییر فونت‌ها
- تنظیم اندازه‌ی فونت برچسب‌های تیک (Tick Labels)

رسم نمودار پیشرفته

مدل‌سازی داده‌های چندبعدی با استفاده از برجستگی‌های سطح سه‌بعدی (3D surface projections)، نقشه‌های حرارتی (heatmaps) و نمودارهای کانتور (contour plots). این شامل استفاده از `matplotlib.animation` برای مصورسازی گذارهای حالت زمانی و جریان‌های داده‌ی پویا می‌شود:

- ایجاد نمودارهای سه‌بعدی
- نمودارهای سطح و سیمی (Surface & Wireframe)
- نقشه‌های حرارتی و نمودارهای کانتور
- انیمیشن‌سازی نمودارها

ذخیره و خروجی نمودارها

ذخیره‌ی مصورسازی‌های خود در فرمت‌های مختلف برای گزارش‌ها و ارائه‌ها:

- ذخیره‌ی نمودارها به صورت PNG، SVG و JPG
- خروجی نمودارها به PDF
- ذخیره‌ی چندین شکل (Figure)

نویسنده : مهران جاوید



00:00:00:00



مصاحبه با دکتر بابک نوری مقدم

عضو هیأت علمی مهندسی کامپیوتر دانشگاه آزاد اسلامی واحد اردبیل
متخصص حوزه هوش مصنوعی و یادگیری ماشین و شبکه های عصبی
از مسیر ورود به کامپیوتر تا آینده هوش مصنوعی و مهندسی کامپیوتر



در این گفت‌وگوی اختصاصی با دکتر بابک نوری مقدم درباره مسیر تحصیلی و حرفه‌ای، تحولات فناوری و آینده مهندسی کامپیوتر به گفت‌وگو پرداختیم. این مصاحبه علاوه بر جنبه‌های تخصصی، شامل تجربه‌ها و توصیه‌هایی برای دانشجویان و علاقه‌مندان به این حوزه بود.

مسیر ورود به دنیای کامپیوتر

دکتر نوری مقدم از علاقه خود به فناوری از دوران کودکی و نخستین تجربه‌هایش با کامپیوتر گفت و توضیح داد که چگونه این علاقه در طول سال‌های تحصیل ادامه پیدا کرد. همچنین درباره انتخاب رشته کامپیوتر و آزادی عمل و تنوع مسیرهای شغلی در این رشته صحبت شد.

چرا هوش مصنوعی؟

یکی از بخش‌های مهم مصاحبه، مسیر حرکت ایشان از حوزه‌های شبکه و امنیت به سمت یادگیری ماشین و هوش مصنوعی بود. او از تأثیر تحصیلات تکمیلی و فعالیت‌های پژوهشی در این تغییر مسیر گفت و هوش مصنوعی را یکی از مهم‌ترین حوزه‌های آینده کامپیوتر دانست.

آینده هوش مصنوعی و AGI

در ادامه درباره تحولات پیش‌روی هوش مصنوعی، افزایش توان پردازشی و توسعه مدل‌های قدرتمندتر صحبت شد. همچنین مفهوم هوش مصنوعی عمومی یا AGI و تفاوت آن با مدل‌های فعلی مورد بحث قرار گرفت.



00:00:00:00





00:00:00:00



هوش مصنوعی و آینده بازار کار

تأثیر AI بر مشاغل حوزه کامپیوتر از موضوعات اصلی این گفت‌وگو بود. دکتر نوری مقدم معتقد است بسیاری از مشاغل لزوماً از بین نمی‌روند، بلکه ماهیت آن‌ها تغییر می‌کند؛ از جمله حرکت از «برنامه‌نویس» به سمت «مهندس نرم‌افزار» و افزایش اهمیت طراحی، مدیریت و درک ساختار نرم‌افزار. چگونه دانشجویان از هوش مصنوعی استفاده کنند؟ یکی از موضوعات مهم مصاحبه، نگرانی درباره وابستگی دانشجویان به ابزارهای هوش مصنوعی بود. پیشنهاد ایشان این است که AI به‌عنوان یک استاد مکمل مورد استفاده قرار گیرد؛ یعنی دانشجو ابتدا یاد بگیرد، مطالعه کند و جست‌وجو انجام دهد و سپس از هوش مصنوعی برای تکمیل یادگیری خود کمک بگیرد.

چالش‌های دانشجویان و اهمیت کنجکاوی

از نگاه دکتر نوری مقدم، یکی از مشکلات دانشجویان این است که گاهی بدون تجربه‌کردن حوزه‌های مختلف، دیر به دنبال پیدا کردن مسیر تخصصی خود می‌روند. او توصیه کرد دانشجویان در سال‌های ابتدایی، حوزه‌های مختلف مانند شبکه، برنامه‌نویسی، وب، سخت‌افزار و رباتیک را تجربه کنند تا علایق و توانایی‌های خود را بهتر بشناسند.

پژوهش، کار تیمی و مهارت‌های نرم

وضعیت پژوهش در حوزه کامپیوتر، ضرورت یادگیری پایه‌ها پیش از ورود جدی به Research، اهمیت کار تیمی و تجربه شخصی دکتر نوری مقدم از چالش‌های همکاری در پروژه‌ها و استارت‌آپ‌ها نیز از دیگر موضوعات مطرح‌شده در این مصاحبه بود.

یک توصیه برای آینده

در پایان، دکتر نوری مقدم بر اهمیت «تلاش، پشتکار و تعهد» تأکید کرد و موفقیت را نتیجه ادامه‌دادن مسیر حتی پس از شکست‌های متعدد دانست. او همچنین یادگیری زبان دوم و توانایی استفاده از مدل‌های هوش مصنوعی را از مهارت‌های مهم برای آینده معرفی کرد.

برای مشاهده نسخه کامل
مصاحبه اسکن کنید یا به
سایت نتوتک مراجعه کنید



00:00:00:00



بهترین ابزارهای هوش مصنوعی به تفکیک نیاز

۱. چت عمومی و استدلال (General Chat & Reasoning)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
ChatGPT	چت عمومی، ایده پردازی و تحلیل متنی	بهترین گزینه برای مکالمات روزمره و حل مسائل عمومی. نسخه پیشرفته برای استدلال عالی است	محدود : سهمیه استفاده روزانه دارد
Claude	درک بافتار طولانی و لحن طبیعی	در نگارش متون طولانی و درک لحن انسانی از ChatGPT روان تر عمل می کند.	محدود : تعداد پیام های رایگان محدود است
DeepSeek	استدلال ریاضی و منطقی قدرتمند	عملکرد استدلالی هم رده مدل های مطرح با سرعت بالا، مناسب برای تحلیل های سنگین.	بدون محدودیت دسترسی آزاد

۲. برنامه نویسی و توسعه نرم افزار (Coding & Software Development)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
Cursor	IDE هوشمند برای توسعه کل پروژه	بهترین ابزار برای باز نویسی و ساخت کامل پروژه؛ کد بیس را به صورت یکپارچه درک می کند.	محدود : تعداد درخواست های هوشمند محدود است
Claude Code	توسعه عامل محور مستقیماً از طریق ترمینال	عالی برای اجرای خودکار دستورات، اجرای تست ها، عیب یابی و مدیریت Git در ترمینال.	محدود : نیازمند اشتراک / سهمیه API
GitHub Copilot	تکمیل خودکار کد حین تایپ	برای پیشنهادهای سریع خط به خط در محیط VS Code بسیار مناسب است.	محدود : نیازمند اشتراک

۳. اتوماسیون و گردش کار (Automation & Workflow)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
n8n	اتوماسیون هوشمند و اتصال عامل های AI	بهترین ابزار برای اتوماسیون؛ قابلیت اجرا روی سرور شخصی، بدون محدودیت و با امنیت بالا برای اتصال AI به دیتابیس.	بدون محدودیت : در صورت نصب شخصی Self-Hosted /
Make	اتوماسیون تصویری و دیداری	محیط بصری بسیار ساده برای اتصال برنامه ها، اما نسبت به n8n انعطاف کمتری در سناریوهای پیچیده دارد.	محدود : تعداد عملیات ماهانه محدود است
Zapier Central	اتوماسیون سریع بدون کدنویسی	ساده ترین راه برای ساخت اکشن های هوشمند سریع با واسطه های آماده.	محدود : تعداد فراخوانی محدود

۴. پژوهش و مقاله‌نویسی (Research & Academic Writing)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
NotebookLM	تحلیل و خلاصه‌سازی اسناد و PDF ها	بهترین ابزار برای مطالعه مقالات؛ فقط بر اساس فایل‌های آپلودشده پاسخ می‌دهد و توهم ندارد.	بدون محدودیت ارائه‌شده توسط گوگل
Perplexity AI	موتور جستجوی پژوهشی و منبع‌یابی	بهترین جایگزین گوگل برای پژوهش؛ پاسخ‌ها را مستقیم با ذکر دقیق منابع علمی ارائه می‌دهد.	محدود جستجوی پیشرفته Pro محدودیت روزانه دارد
Elicit	مرور ادبیات تحقیق و استخراج داده	تخصصی‌ترین ابزار برای یافتن مقالات معتبر علمی و استخراج خودکار روش‌شناسی.	محدود اعتبار اولیه محدود

۵. تولید و ویرایش تصویر (Image Generation & Editing)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
Midjourney	بالاترین کیفیت بصری و خروجی هنری	بهترین کیفیت تصویر، نورپردازی و جزئیات واقع‌گرایانه در صنعت.	محدود نیازمند اشتراک
Flux.1	تولید متن دقیق روی تصویر و آناتومی	بهترین مدل متن‌باز؛ در نگارش حروف روی تصویر و رندر جزئیات دقیق‌تر عمل می‌کند.	بدون محدودیت متن‌باز با قابلیت اجرای محلی
DALL-E 3	درک دقیق دستورات متنی پیچیده	درون ChatGPT فعال است؛ فهم پرامپت بالایی دارد اما کیفیت هنری آن پایه Midjourney نیست.	محدود وابسته به سهمیه ChatGPT

۶. تولید ویدئو و آواتار (Video & Avatar Generation)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
Runway Gen-3	کنترل سینمایی بر حرکت دوربین و سوزه	بهترین ابزار برای ساخت تیزر و افکت‌های حرفه‌ای با کنترل دقیق بر حرکات.	محدود اعتبار اولیه محدود
Luma Dream Machine	تولید ویدیوهای پویا و سریع	سرعت رندر بالا و ساخت حرکت‌های سه‌بعدی واقع‌گرایانه از روی عکس ثابت.	محدود تعداد رندر روزانه محدود
HeyGen	ساخت آواتار سخنگوی واقع‌گرایانه	بهترین ابزار ساخت ویدیوهای آموزشی و مجری مجازی با همگام‌سازی لب و ترجمه ویدیو.	محدود اعتبار اولیه محدود

۷. تولید صدا، موسیقی و ویرایش صوت (Audio & Music)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
ElevenLabs	طبیعی‌ترین همانندسازی و گویندگی صدا	بهترین ابزار گویندگی و کلون صدا با حفظ لحن و احساسات (پشتیبانی عالی از زبان فارسی).	محدود تعداد کاراکتر رایگان محدود
Suno AI	آهنگسازی و تولید کامل ساختار موسیقی	بهترین ابزار موسیقی؛ شعر، خوانندگی و سازبندی را در تمام سبک‌ها تولید می‌کند.	محدود اعتبار روزانه محدود
Adobe Express (Enhance Speech)	حذف نویز و تمیز کردن صدای پادکست	تبدیل صدای ضبط‌شده با موبایل یا کیفیت پایین به کیفیت استودیویی با یک کلیک.	بدون محدودیت بخش اصلاح صوت رایگان است

۸. طراحی گرافیک، UI/UX و سایت‌ساز (Design & Web)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
Canva (Magic Studio)	طراحی سریع پوستر و محتوای تصویری	بهترین ابزار برای کارهای سریع گرافیکی به همراه قابلیت‌های اصلاح تصویر.	محدود امکانات حرفه‌ای نیازمند اشتراک است
Figma AI	طراحی رابط کاربری (UI/UX)	بهترین ابزار برای طراحان اپ و سایت جهت ساخت خودکار کامپوننت‌ها با پرامپت.	محدود سهمیه ابزارهای هوشمند محدود است
Framer AI	ساخت سریع وب‌سایت با پرامپت متنی	ساخت وب‌سایت‌های ریسپانسیو و انیمیت شده تنها با نوشتن توصیف متنی.	محدود امکانات پیشرفته پولی است

۹. کسب‌وکار، مدیریت و تحلیل داده (Business & Data)

ابزار	کاربرد و نقطه قوت اصلی	بررسی و مقایسه	وضعیت محدودیت دسترسی
Julius AI	تحلیل فایل‌های اکسل و رسم نمودار	دریافت فایل داده، تحلیل آمار به زبان ساده و رسم خودکار نمودارهای تخصصی.	محدود تعداد پیام‌های ماهانه محدود
Otter.ai	پیاده‌سازی متن و خلاصه جلسات	شرکت در جلسات آنلاین، تبدیل گفتار به متن و استخراج نکات کلیدی.	محدود محدودیت دقایق ضبط در ماه
DeepL	ترجمه فوق‌العاده دقیق و طبیعی اسناد	دقیق‌ترین مترجم برای متون تخصصی، حقوقی و اداری؛ روان‌تر از Google Translate.	محدود محدودیت تعداد کاراکتر در نسخه رایگان

نویسنده: کیمیا رهنما آذر و سحر سیدایمانی

Home

Repos

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

Search GitHub...

Neotech

Build · Learn · Grow

Follow

Overview

Repositories

Projects

Packages

Stars

Perlin Noise: الگوریتمی برای تولید رویه‌ای

نویسنده: مهران جاوید

نویز پرلین یک الگوریتم محبوب تولید رویه‌ای (procedural generation) است که توسط کن پرلین ابداع شده است. از این الگوریتم می‌توان برای تولید رویه‌ای چیزهایی مانند بافت‌ها و سطوح زمین استفاده کرد؛ به این معنا که نیازی به ساخت دستی آن‌ها توسط هنرمند یا طراح نیست. این الگوریتم می‌تواند یک یا چند بُعد داشته باشد که در واقع تعداد ورودی‌های آن را مشخص می‌کند. در این مقاله، من از دو بُعد استفاده می‌کنم چون تجسم‌پذیری آن نسبت به سه بُعد آسان‌تر است.

همچنین سردرگمی زیادی در مورد این که نویز پرلین چیست و چیست، وجود دارد. اغلب با نویز مقدار (value noise) و نویز سیمپلکس (simplex noise) اشتباه گرفته می‌شود. در مجموع چهار نوع نویز مشابه وجود دارند که اغلب با یکدیگر اشتباه گرفته می‌شوند: نویز پرلین کلاسیک، نویز پرلین بهبودیافته، نویز سیمپلکس، و نویز مقدار. نویز پرلین بهبودیافته، نسخه‌ی بهبودیافته‌ی نویز پرلین کلاسیک است. نویز سیمپلکس متفاوت است، اما آن هم توسط کن پرلین ساخته شده است. نویز مقدار نیز متفاوت است. یک قاعده‌ی سرانگشتی این است که اگر الگوریتم نویز از یک مولد اعداد (شبه-) تصادفی استفاده می‌کند، احتمالاً نویز مقدار است. این مقاله درباره‌ی نویز پرلین بهبودیافته است.

ابتدا، نحوه‌ی استفاده از آن. الگوریتم به عنوان ورودی، تعداد مشخصی پارامتر اعشاری (با توجه به تعداد ابعاد) دریافت می‌کند و مقداری را در بازه‌ای مشخص برمی‌گرداند (برای نویز پرلین، معمولاً گفته می‌شود که این بازه بین -1/۰ تا 1/۰+ است، اما در واقعیت کمی متفاوت است). فرض کنیم در دو بُعد باشد، بنابراین دو پارامتر ورودی دارد x و y . حال، x و y می‌توانند هر چیزی باشند، اما معمولاً یک موقعیت (position) را نشان می‌دهند. برای تولید یک بافت (texture)، x و y مختصات پیکسل‌های موجود در آن بافت خواهند بود (ضرب شده در عدد کوچکی به نام فرکانس که در انتها به آن خواهیم پرداخت). بنابراین برای تولید بافت، از میان تمام پیکسل‌های موجود در آن حلقه می‌زنیم، تابع نویز پرلین را برای هر کدام فراخوانی می‌کنیم و بر اساس مقدار بازگشتی، تعیین می‌کنیم که آن پیکسل چه رنگی داشته باشد.

یک نمونه پیاده‌سازی به شکل زیر خواهد بود:

```
Color pixels[500][500];

for (int y = 0; y < 500; y++) {
    for (int x = 0; x < 500; x++) {
        // Noise2D generally returns a value approximately in the range [-1.0, 1.0]
        float n = Noise2D(x*0.01, y*0.01);

        // Transform the range to [0.0, 1.0], supposing that the range of Noise2D is [-1.0, 1.0]
        n += 1.0;
        n /= 2.0;

        int c = Math.round(255*n);
        pixels[y][x] = Color(c, c, c);
    }
}
```

Ⓜ

Home

📁

Repos

❗

Issues

🔗

Pull requests

▶

Actions

📁

Projects

📖

Wiki

🛡️

Security

📈

Insights

⚙️

Settings

🔍 Search GitHub...

🔔

+

👤



Neotech

Build · Learn · Grow

Follow

...

Overview

Repositories

Projects

Packages

Stars

اجرای این کد تصویری مشابه تصویر زیر را به همراه خواهد داشت:

همان‌طور که می‌بینید، هر پیکسل فقط یک رنگ تصادفی ندارد؛ در عوض، از پیکسلی به پیکسل دیگر تغییراتی هموار و پیوسته دارند و بافت نهایی ظاهری تصادفی ندارد. این ویژگی به دلیل خاصیت نویز پرلین (و سایر انواع نویز) است که اگر دو ورودی به یکدیگر نزدیک باشند مثلاً ۲.۵، ۳.۱۱ و ۲.۵۱، خروجی تابع نویز نیز به یکدیگر نزدیک خواهد بود.

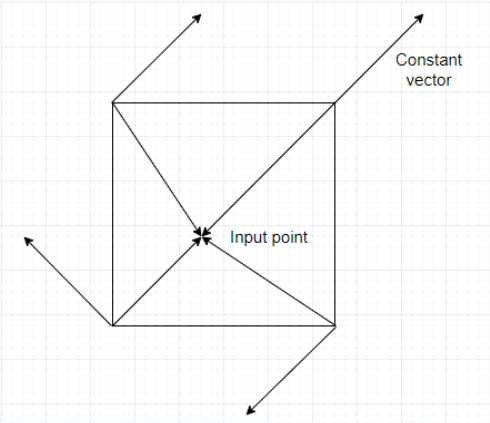
حال، سازوکار آن به چه صورت است؟

ابتدا توضیحی مختصر ارائه می‌دهم و سپس به تفصیل به آن خواهیم پرداخت:

ورودی‌ها روی یک شبکه‌ی صحیح (integer grid) در نظر گرفته می‌شوند (شکل ۲ را ببینید). هر ورودی اعشاری درون یکی از مربع‌های این شبکه قرار می‌گیرد. برای هر یک از ۴ گوشه‌ی آن مربع، مقداری تولید می‌کنیم. سپس بین آن ۴ مقدار درونیابی (interpolate) می‌کنیم و به نتیجه‌ی نهایی می‌رسیم. تفاوت بین نویز پرلین و نویز مقدار (value noise) در نحوه‌ی به‌دست‌آوردن آن ۴ مقدار است. در حالی که نویز مقدار از یک مولد اعداد شبه‌تصادفی استفاده می‌کند، نویز پرلین از ضرب داخلی (dot product) بین دو بردار بهره می‌برد.

(0,2)	(1,2)	(2,2)
(0,1)	(1,1)	(2,1)
(0,0)	(1,0)	(2,0)

Input point
(1.67, 1.33)



اول، بردار اشاره‌گر از نقطه‌ی شبکه (گوشه‌ها) به نقطه‌ی ورودی است. بردار دیگر، بردار ثابتی است که به هر نقطه‌ی شبکه نسبت داده شده است (شکل ۳ را ببینید). این بردار باید همواره برای یک نقطه‌ی شبکه‌ی مشخص، یکسان باشد، اما در صورت تغییر seed الگوریتم، می‌تواند تغییر کند (چگونگی آن را در ادامه خواهیم دید).

در اغلب پیاده‌سازی‌های نویز پرلین، خروجی نویز در فواصل مضربی از ۲۵۶ که در ادامه آن را با w نشان می‌دهیم (دچار دورزدن (wrap) شده و تکرار می‌شود. علت این رفتار، استفاده از ساختاری به نام جدول جایگشت (permutation table) برای تخصیص بردارهای ثابت به نقاط شبکه است.

⌵

🐱

Home

📁

Repos

❗

Issues

🔗

Pull requests

▶

Actions

📁

Projects

📖

Wiki

🛡️

Security

📈

Insights

⚙️

Settings

🐱

🔍 Search GitHub...

🔔

+

👤

🐱

Neotech

Build · Learn · Grow

Follow

⋮

📖 Overview

📁 Repositories

📁 Projects

📁 Packages

★ Stars

```
// Suppose x, y and z are the float input
const X = Math.floor(x) & 255; // Used Later
const Y = Math.floor(y) & 255; // Used Later
const xf = x-Math.floor(x);
const yf = y-Math.floor(y);

const topRight = new Vector2(xf-1.0, yf-1.0);
const topLeft = new Vector2(xf, yf-1.0);
const bottomRight = new Vector2(xf-1.0, yf);
const bottomLeft = new Vector2(xf, yf);
```

جدول جایگشت، آرایه‌ای به طول W است که شامل تمامی اعداد صحیح بین 0 تا $W-1$ به صورت درهم‌ریخته (یعنی یک جایگشت) می‌باشد. اندیس‌گذاری در این آرایه (مقدار درون کروشه‌ها) با استفاده از X یا Y (یا مقداری نزدیک به آن‌ها) صورت می‌گیرد، بنابراین مقدار اندیس باید همواره کمتر از 256 باشد. علت دورزدن نویز این است که برای نمونه، اگر ورودی X برابر 256 باشد، مقدار X برابر صفر خواهد شد. این صفر برای اندیس‌گذاری در جدول جایگشت و تولید بردار تصادفی به کار می‌رود. از آنجا که X در تمام مضارب 256 مقداری برابر صفر دارد، بردار تصادفی تولیدشده نیز در تمام این نقاط یکسان خواهد بود و در نتیجه نویز تکرار می‌شود. در صورت نیاز، می‌توان اندازه‌ی جدول جایگشت را افزایش داد (برای نمونه به 512) که در این صورت نویز در مضارب 512 تکرار خواهد شد.

در گام نخست، جدول جایگشت را ساخته و آن را به صورت تصادفی درهم‌می‌ریزیم. در ادامه، قطعه‌کد مربوطه ارائه شده و توضیحات آن بلافاصله پس از آن خواهد آمد:

```
// Create an array (our permutation table) with the values 0 to 255 in order
const permutation = [];
for(let i = 0; i < 256; i++) {
    permutation[i] = i;
}

// Shuffle it
permutation = Shuffle(permutation);
```

یک نمونه از تابع درهم‌ریزی مورد استفاده، در قطعه‌کد کامل ارائه‌شده در انتهای مقاله قابل مشاهده است.

گام بعدی، استخراج یک مقدار از جدول جایگشت برای هر یک از گوشه‌هاست. اما در اینجا یک قید اساسی وجود دارد: هر گوشه‌ی شبکه، مستقل از اینکه نقطه‌ی ورودی در کدام یک از چهار خانه‌ی شبکه‌ای که آن گوشه را شامل می‌شوند قرار داشته باشد، باید همواره مقدار ثابتی را بازگرداند. به عنوان نمونه، اگر گوشه‌ی بالا-راست خانه‌ی شبکه‌ی $(0, 0)$ مقدار 42 داشته باشد، گوشه‌ی بالا-چپ خانه‌ی شبکه‌ی $(0, 1)$ نیز باید دقیقاً همان مقدار 42 را داشته باشد. از آنجا که این دو، در واقع یک نقطه‌ی شبکه‌ی مشترک هستند، مقدار آن‌ها بدون توجه به خانه‌ی شبکه‌ای که از آن محاسبه می‌شود، باید یکسان باقی بماند:



Home



Repos



Issues



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings

**Neotech**

Build · Learn · Grow

Follow



Overview

Repositories

Projects

Packages

Stars

```
// Select a value in the array for each of the 4 corners
const valueTopRight = P[P[X+1]+Y+1];
const valueTopLeft = P[P[X]+Y+1];
const valueBottomRight = P[P[X+1]+Y];
const valueBottomLeft = P[P[X]+Y];
```

شیوه‌ی انتخاب مقادیر برای گوشه‌ها در قطعه‌کد فوق، قید اشاره‌شده را به‌خوبی تأمین می‌کند. به‌عنوان نمونه، در خانه‌ی شبکه‌ای (۰, ۰)، مقدار $valueBottomRight$ از عبارت $P[P[0+1]+0] = P[P[1]+0]$ حاصل می‌شود. در مقابل، در خانه‌ی شبکه‌ای (۱, ۰)، مقدار $valueBottomLeft$ برابر با $P[P[1]+0]$ خواهد بود. همان‌طور که مشاهده می‌شود، دو مقدار $valueBottomLeft$ و $valueBottomRight$ با یکدیگر برابر هستند؛ بنابراین محدودیت موردنظر به‌درستی برقرار شده است.

به‌منظور دورزدن (wrap) نویز در هر مضربی از ۲۵۶، لازم است جدول جایگشت را دوبار برابر کنیم. در شرایطی که در حال محاسبه‌ی $P[X+1]$ هستیم و X برابر ۲۵۵ است) بنابراین $X+1$ برابر ۲۵۶ خواهد بود)، در صورتی که آرایه را دوبار برابر نکرده باشیم، با خطای سرریز (overflow) مواجه خواهیم شد، زیرا بیشینه‌ی اندیس مجاز در یک آرایه‌ی ۲۵۶ تایی، ۲۵۵ است. اما نکته‌ی حائز اهمیت این است که ترتیب انجام این دو عملیات بسیار مهم است: نباید ابتدا آرایه را دوبار برابر کنیم و سپس آن را درهم‌بریزیم؛ بلکه باید ابتدا درهم‌ریزی را انجام داده و سپس آن را دوبار برابر کنیم. دلیل این امر در مثال $P[X+1] = P[256]$ به‌وضوح دیده می‌شود: ما نیاز داریم که $P[X+1]$ دقیقاً همان مقدار $P[0]$ را داشته باشد تا پدیده‌ی دورزدن به‌درستی رخ دهد.

در این مرحله، نوبت به استخراج بردارهای ثابت می‌رسد. در پیاده‌سازی اصلی کن پرلین، تابعی با نام «grad» به‌کار رفته بود که مستقیماً ضرب داخلی (dot product) را برای هر گوشه محاسبه می‌کرد. اما در اینجا رویکرد ساده‌تری را در پیش می‌گیریم: تابعی ایجاد می‌کنیم که صرفاً بردار ثابت متناظر با هر مقدار از جدول جایگشت را بازگرداند و محاسبه‌ی ضرب داخلی را به مرحله‌ی بعد موکول کنیم.

به‌علاوه، به‌دلیل سهولت در تولید، این بردارهای ثابت محدود به چهار بردار زیر هستند:

، $(1.0, 1.0)$ ، $(1.0, -1.0)$ ، $(-1.0, 1.0)$ ، $(-1.0, -1.0)$ به‌منظور استخراج بردارهای ثابت از روی یک مقدار برگرفته از جدول جایگشت، می‌توان از روش زیر استفاده کرد:

```
function GetConstantVector(v) {
    // v is the value from the permutation table
    const h = v & 3;
    if(h === 0)
        return new Vector2(1.0, 1.0);
    else if(h === 1)
        return new Vector2(-1.0, 1.0);
    else if(h === 2)
        return new Vector2(-1.0, -1.0);
    else
        return new Vector2(1.0, -1.0);
}
```

 Home

Repos



Issues



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings



Neotech

Build · Learn · Grow

Follow



Overview

Repositories

 Projects Packages

☆ Stars

با توجه به اینکه ۷ در بازه ۰ تا ۲۵۵ قرار دارد و تعداد بردارهای ممکن ۴ عدد است، با اعمال عملگر & 3 (که معادل % ۴ است)، چهار مقدار ممکن برای h (۰، ۱، ۲، ۳) حاصل می‌شود. بر اساس هر یک از این مقادیر، بردار ثابت متناظر بازگردانده می‌شود.

پس از این مرحله، نوبت به محاسبه ضرب داخلی (dot product) می‌رسد:

```
const dotTopRight = topRight.dot(GetConstantVector(valueTopRight));
const dotTopLeft = topLeft.dot(GetConstantVector(valueTopLeft));
const dotBottomRight = bottomRight.dot(GetConstantVector(valueBottomRight));
const dotBottomLeft = bottomLeft.dot(GetConstantVector(valueBottomLeft));
```

پس از محاسبه‌ی ضرب داخلی برای هر یک از چهار گوشه، گام بعدی ترکیب این مقادیر به‌گونه‌ای است که یک مقدار خروجی واحد حاصل شود. این کار با استفاده از درون‌یابی (interpolation) انجام می‌پذیرد. درون‌یابی روشی است که به‌کمک آن می‌توان مقداری را که بین دو مقدار مشخص (a_1 و a_2) قرار دارد، بر اساس پارامتری به‌نام t در بازه‌ی 0.0 تا 1.0 (که معادل درصدی بین 0% تا 100% است) تعیین کرد. به‌عنوان نمونه، اگر $a_1=10$ ، $a_2=20$ و $0.5t=50\%$ باشد، مقدار درون‌یابی‌شده برابر 15 خواهد بود، زیرا نقطه‌ی میانی بین دو مقدار را نشان می‌دهد. در مثالی دیگر، اگر $a_1=50$ ، $a_2=100$ و $0.4t=40\%$ باشد، مقدار درون‌یابی‌شده در 70% مسیر بین 50 و 100 قرار می‌گیرد که برابر 70 است. این روش با عنوان درون‌یابی خطی (linear interpolation) شناخته می‌شود، زیرا مقادیر حاصل بر روی یک خط مستقیم قرار می‌گیرند.

پس از محاسبه‌ی چهار مقدار ضرب داخلی برای چهار گوشه، نوبت به درونیابی (interpolation) آن‌ها می‌رسد. از آنجا که در هر مرحله تنها می‌توان دو مقدار را با یکدیگر درونیابی کرد، فرایند به‌صورت زیر انجام می‌شود: نخست، مقادیر مربوط به گوشه‌های بالا-چپ و پایین-چپ با هم درونیابی می‌شوند تا مقدار v_1 حاصل گردد. سپس، مقادیر گوشه‌های بالا-راست و پایین-راست به‌همین ترتیب درونیابی شده و مقدار v_2 به‌دست می‌آید. در گام نهایی، بین v_1 و v_2 درونیابی انجام می‌شود تا مقدار نهایی خروجی حاصل شود. این مقدار نهایی، همان مقداری است که تابع نویز پرلین باید به‌عنوان خروجی بازگرداند.

در ادامه، پیاده‌سازی تابع درونیابی خطی (linear interpolation) که به اختصار lerp نامیده می‌شود، ارائه شده است:

```
function Lerp(t, a1, a2) {  
    return a1 + t*(a2-a1);  
}
```

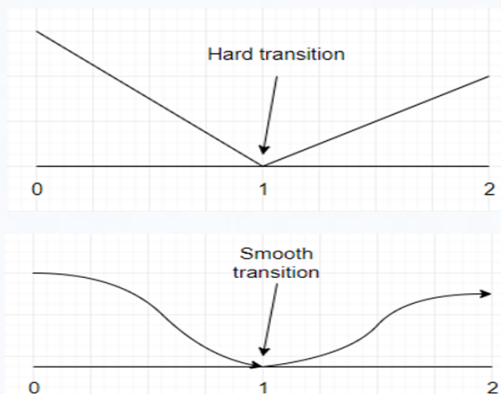
ما می‌توانستیم از درون‌یابی خطی استفاده کنیم،

اما این کار نتایج چندان خوبی به همراه نخواهد داشت

، زیرا ظاهری غیرطبیعی ایجاد می‌کند، همان‌طور که در

این تصویر که درون یابی خطی یک بعدی را نشان می‌دهد،

قابل مشاهده است:





Home



Repos



Issues



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings



Neotech

Build · Learn · Grow

Follow



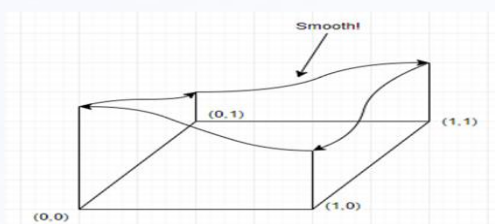
Overview

Repositories

Projects

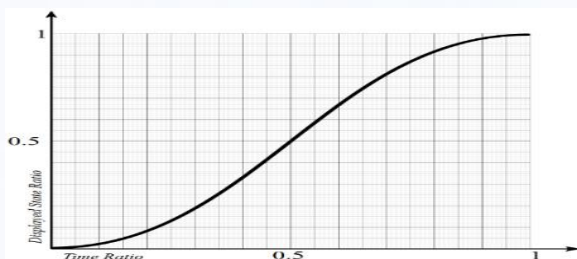
Packages

Stars



همان‌طور که در شکل مشخص است، گذار از مقادیر کمتر از ۱ به مقادیر بیشتر از ۱ به‌صورت ناگهانی و بدون همواری صورت می‌گیرد. در حالی که هدف ما دستیابی به تغییری هموارتر و طبیعی‌تر است، مانند نمونه‌ی زیر:

در رویکرد درونی‌یابی خطی، از x_f به‌عنوان پارامتر درونی‌یابی (t) استفاده می‌شود. اما در اینجا، x_f و y_f را به ترتیب به u و v تبدیل می‌کنیم. این تبدیل به‌گونه‌ای طراحی شده است که برای مقادیر t در بازه‌ی (۰.۵ تا ۰.۵)، مقدار حاصل اندکی کوچک‌تر از t باشد (با کران پایین ۰.۵). همچنین برای مقادیر t در بازه‌ی (۰.۵ تا ۱.۰)، مقدار حاصل اندکی بزرگ‌تر از t خواهد بود (با کران بالای ۱.۰). در نقطه‌ی ۰.۵، مقدار تبدیل‌شده دقیقاً برابر ۰.۵ باقی می‌ماند. این رویکرد منجر به ایجاد یک انتقال منحنی‌شکل (curvy transition) می‌شود که در شکل‌های ۵ و ۶ به‌نمایش درآمده است.



به‌منظور تحقق این هدف، از ابزاری به‌نام منحنی نرم‌کننده (ease curve) استفاده می‌کنیم. این منحنی در واقع یک تابع ریاضی است که نمودار آن به‌صورت زیر است:

[شکل ۷] نمایی از یک منحنی نرم‌کننده

با دقت در نمودار فوق، مشاهده می‌شود که برای مقادیر ورودی (x_f یا y_f)، در محور افقی) در بازه‌ی ۰.۵ تا ۰.۵، خروجی (u یا v)، در محور عمودی) اندکی به ۰.۵ نزدیک‌تر است. به‌همین‌ترتیب، برای ورودی‌های بین ۰.۵ تا ۱.۰، خروجی اندکی به ۱.۰ میل می‌کند. در نقطه‌ی $0.5x=$ ، خروجی دقیقاً برابر ۰.۵ خواهد بود. این رفتار، دقیقاً همان چیزی است که برای دستیابی به انتقالی هموار و طبیعی به آن نیاز داریم.

تابع نرم‌کننده‌ای (ease function) که در نمودار فوق نمایش داده شده است، همان تابعی است که کن پرلین در پیاده‌سازی اصلی خود از نویز پرلین به‌کار برده است. این تابع با معادله‌ی $6t^4 + 15t^3 - 6t^2$ تعریف می‌شود و با عنوان تابع محوشونده (fade function) نیز شناخته می‌شود. پیاده‌سازی این تابع در کد به‌صورت زیر خواهد بود.

```
// Unoptimized version
function Fade(t) {
    return 6*t*t*t*t*t - 15*t*t*t*t + 10*t*t*t;
}

// Optimized version (less multiplications)
function Fade(t) {
    return ((6*t - 15)*t + 10)*t*t*t;
}
```



Search GitHub...



Home



Repos



Issues



Pull requests



Actions



Projects



Wiki



Security



Insights



Settings



Neotech

Build · Learn · Grow

Follow



Overview

Repositories

Projects

Packages

Stars

در این مرحله، تنها کافی است درون یابی خطی را مطابق با روش پیش تر شرح داده شده، با این تفاوت که اکنون از u و v به عنوان پارامترهای درون یابی (t) استفاده می کنیم، انجام دهیم. پیاده سازی این بخش در کد به صورت زیر است:

```
const u = Fade(xf);
const v = Fade(yf);
const result = Lerp(u,
    Lerp(v, dotBottomLeft, dotTopLeft),
    Lerp(v, dotBottomRight, dotTopRight)
);
```

بدین ترتیب، پیاده سازی نویز پرلین به پایان می رسد. کد کامل این الگوریتم در ادامه ارائه شده است:

```
class Vector2 {
    constructor(x, y) {
        this.x = x;
        this.y = y;
    }

    dot(other) {
        return this.x*other.x + this.y*other.y;
    }
}

function Shuffle(arrayToShuffle) {
    for(let e = arrayToShuffle.length-1; e > 0; e--) {
        const index = Math.round(Math.random()*(e-1));
        const temp = arrayToShuffle[e];
```

```
        arrayToShuffle[e] = arrayToShuffle[index];
        arrayToShuffle[index] = temp;
    }
}

function MakePermutation() {
    const permutation = [];
    for(let i = 0; i < 256; i++) {
        permutation.push(i);
    }

    Shuffle(permutation);

    for(let i = 0; i < 256; i++) {
        permutation.push(permutation[i]);
    }

    return permutation;
}
```

⌵

🐱

Home

📁

Repos

❗

Issues

🔗

Pull requests

▶

Actions

📁

Projects

📖

Wiki

🛡️

Security

📈

Insights

⚙️

Settings

🐱

🔍 Search GitHub...

🔔

+

👤

🐱

Neotech

Build · Learn · Grow

Follow

⋮

📖 Overview

📁 Repositories

📁 Projects

📁 Packages

★ Stars

```
const Permutation = MakePermutation();

Function GetConstantVector(v) {
    // v is the value from the permutation table
    const h = v & 3;
    if(h == 0)
        return new Vector2(1.0, 1.0);
    else if(h == 1)
        return new Vector2(-1.0, 1.0);
    else if(h == 2)
        return new Vector2(-1.0, -1.0);
    else
        return new Vector2(1.0, -1.0);
}

Function Fade(t) {
    return ((6*t - 15)*t + 10)*t*t*t;
}

Function Lerp(t, a1, a2) {
    return a1 + t*(a2-a1);
}

Function Noise2D(x, y) {
    const X = Math.floor(x) & 255;
    const Y = Math.floor(y) & 255;

    const xf = x-Math.floor(x);
    const yf = y-Math.floor(y);

    const topRight = new Vector2(xf-1.0, yf-1.0);
    const topLeft = new Vector2(xf, yf-1.0);
    const bottomRight = new Vector2(xf-1.0, yf);
    const bottomLeft = new Vector2(xf, yf);

    // Select a value from the permutation array for each of the 4 corners
    const valueTopRight = Permutation[Permutation[X+1]+Y+1];
    const valueTopLeft = Permutation[Permutation[X]+Y+1];

    const valueBottomRight = Permutation[Permutation[X+1]+Y];
    const valueBottomLeft = Permutation[Permutation[X]+Y];

    const dotTopRight = topRight.dot(GetConstantVector(valueTopRight));
    const dotTopLeft = topLeft.dot(GetConstantVector(valueTopLeft));
    const dotBottomRight =
        bottomRight.dot(GetConstantVector(valueBottomRight));
    const dotBottomLeft =
        bottomLeft.dot(GetConstantVector(valueBottomLeft));

    const u = Fade(xf);
    const v = Fade(yf);

    return Lerp(
        Lerp(v, dotBottomLeft, dotTopLeft),
        Lerp(v, dotBottomRight, dotTopRight)
    );
}
```

بدین ترتیب، نویز پرلین را به طور خلاصه مرور کردیم. این الگوریتم کاربردهای گسترده‌ای دارد و می‌توان از آن برای تولید طیف وسیعی از محتواها، از زنجیره‌های کوهستانی گرفته تا نقشه‌های ارتفاع، بهره برد.

NEOTECH

پرسودترین زیرشاخه‌های علوم کامپیوتر

از ابر و داده تا امنیت و هوش مصنوعی؛ نگاهی به سختی، نیاز به برنامه‌نویسی،
بازار کار ایران و جهان، دورکاری و درآمد هشت مسیر شغلی داغ دنیای فناوری.

مهندسی داده

دواپس

مهندسی ابر

هوش مصنوعی و یادگیری ماشین

امنیت تهاجمی و سایبری

بک‌اند و فول‌استک

مهندسی قابلیت‌اطمینان سایت

بلاک‌چین و وب۳

نویسنده: محمد مهدی عباس‌زاده

DevOps Engineer

مهندسی ابر

Cloud Engineering



طراحی، ساخت و مدیریت زیرساخت در حال ابر

4/5

سختی یادگیری



3/5

برنامه‌نویسی موردنیاز



5/5

امکان دورکاری



5/5

تقاضای بازار



مهندسی ابر مسئول طراحی و نگهداری زیرساخت‌هایی است که روی پلتفرم‌هایی مانند Google AWS, Azure, GCCE و ناهای داخلی ArvanCloud اجرا می‌شوند. این حوزه ترکیبی از شبکه، امنیت، اتوماسیون و معماری زیرافزار است و چون تقریباً هر کسب‌وکاری امروزه به ابر وابسته شده، تقاضا برای آن هرگز افت نکرده است.

چرا این حوزه سودآور است

- تقریباً هیچ شرکت بزرگی بدون زیرساخت ابری کار نمی‌کند
- گروه‌دهنده‌های بزرگ مثل Azure و AWS سرمایه‌گذاری‌های کلان دارند
- زیرساخت‌های Infrastructure as Code و Kubernetes دائماً در حال رشدند

بازار کار جهانی

بنگی، زیرساخت‌های، شرکت‌های یرمیوب در دنیا شرکت‌های لامریکایی و اروپایی، فرصت‌های شغلی متنوع با فرصت رشد، جذب استعداد.

= ۳۰۰۰۰ تا ۶۰۰۰ دلار سالیانه
(فرانسی ۵۰ تا ۹۰ هزار در ساعت)

بازار کار در ایران

تقاضا رو به رشد در استارت‌آپ‌ها، بانک‌ها و شرکت‌های لوفتامینی اکثر وهمچنین پروژه‌های احرری داخلی احتیاجه می‌شود.

= ۲۵ تا ۶۰ میلیون تومان در ماه
(سطح مهارت تا ۳ ساله)

دواپس

DevOps Engineering



پل میان توسعه و عملیات، اتوماسیون از صفر تا استقرار

4/5

سختی یادگیری



3/5

برنامه‌نویسی موردنیاز



5/5

امکان دورکاری



5/5

تقاضای بازار



مهندسی دواپس فرآیند ساخت، تست و استقرار نرم‌افزار را خودکار می‌کند تا تیم‌های توسعه سریع‌تر و پایدارتر منتشر کنند. تسلط بر CI/CD، کانیتیرها (مکتبنس Kubernetes)، مانیتورینگ، مستدسازی این یخش است، چون هر تیم برم‌افزاری به سرعت استقرار نیاز دارد، این حوزه یکی از پرفرطرفارترین بازارهای کار فنی است.

چرا این حوزه سودآور است

- هر شرکت با تیم توسعه، نیاز به دواپس نیاز پیدا می‌کند
- مهارت‌های آن (CI/CD, Kubernetes) در همه صنایع قابل انتقال است
- بسیاری از نقش‌ها کاملاً ریموت و آزاد از محدودیت جغرافیایی هستند

بازار کار جهانی

تقاضای بالا در شرکت‌های بزرگ، استارت‌آپ‌ها و پروژه‌های ابرری و فرصت‌های بین‌المللی.

= ۴۰ تا ۹۰ هزار دلار سالیانه
(فرانسی ۳۵ تا ۱۰۰ دلار در ساعت)

بازار کار در ایران

از زیرساخت‌های سازمانی، شرکت‌های آموزشی و استارت‌آپ‌ها، خدمات و محصولات، به وجود آمده.

= ۲۵ تا ۷۰ میلیون تومان در ماه
(سطح مهارت ۱ تا ۳ ساله)

مهندسی داده

Data Engineering



ساخت خط لوله‌هایی که داده خام را به دارایی تبدیل می‌کنند.

4 / 5

سختی یادگیری



4 / 5

برنامه‌نویسی موردنیاز



5 / 5

امکان دورکاری



5 / 5

تقاضای بازار



مهندسی داده مسئول جمع‌آوری، پردازش و ذخیره حجم عظیمی از داده برای مصرف، تحلیل و یادگیری ماشین است. کار با ابزارهایی مثل Apache, Spark, Airflow, Kafka و ابزارهای، ذخیره داده مانند فمیل، پذیر، فضای ابری، همچنین آشنایی با معماری‌های توزیع شده و مقیاس پذیر همیشه شده.

چرا این حوزه سودآور است

- بدون داشتن تجربه، برنامه‌نویسی می‌توانی به عنوان پایه کار کنی.
- مهارت‌های SQL و پاییه آن با نسبتاً قابل یادگیری، به دست می‌آیند.
- شرکت‌های بزرگ و استارت‌آپ‌ها به دنبال جذب این افراد دارند.

بازار کار جهانی

با رشد داده‌های، حجم، تقاضا برای مهندسين داده در شرکت‌های بزرگ، فناوری، ارزانه پادهنده‌های ابری و استارت‌آپ‌ها در حال افزایش است.

≈ ۴۰ تا ۸۰ هزار دلار سالیانه (فرانس: ۶۰ تا ۹۰ هزار در ساعت)

بازار کار در ایران

با رشد صنعت داده و گسترش سرویس‌های آنلاین، شرکت‌های فناوری و بانک‌ها به مهندسين داده به صورت جدی نیاز دارند.

≈ ۲۵ تا ۶۰ میلیون تومان در ماه (سطح مهارت تا ارشد)

امنیت تهاجمی و سایبری

Cybersecurity / Offensive Security



فکر کردن مثل مهاجم برای محافظت از سازمان

5 / 5

سختی یادگیری



3 / 5

برنامه‌نویسی موردنیاز



3 / 5

امکان دورکاری



5 / 5

تقاضای بازار



متخصصان امنیت تهاجمی (تست نفوذ، تیم قرمز) نقاط ضعف سیستم‌ها را پیش از سوءاستفاده مهاجمان واقعی پیدا می‌کنند. این حوزه نیاز به درک عمیق از شبکه، سیستم‌عامل، وب و اسکریپت‌نویسی دارد. با افزایش حملات سایبری، در سطح جهانی، سازمان‌ها نیروی انسانی را به فیسال بیشتر می‌کنند.

چرا این حوزه سودآور است

- افزایش مداوم حملات سایبری، توجه به امنیت را در همه صنایع بالا برده.
- گواهی‌نامه‌هایی مثل OSCP مستقیماً باعث جهش حقوقی می‌شوند.
- برنامه‌های Bug Bounty درآمد جانبی قابل توجهی ایجاد می‌کنند.

بازار کار جهانی

با گسترش تهدیدات سایبری، متخصصان امنیت در شرکت‌های بزرگ، سازمان‌ها و استارت‌آپ‌ها، مورد تقاضا هستند.

≈ ۱۰۰ تا ۲۰۰ هزار دلار سالیانه (فرانس: ۳۵ تا ۱۰۰ دلار در ساعت)

بازار کار در ایران

با افزایش حملات سایبری، نیاز به متخصصان امنیت و تست نفوذ در سازمان‌ها، بانک‌ها و شرکت‌های فناوری رو به رشد است.

≈ ۲۵ تا ۵۵ میلیون تومان در ماه

هوش مصنوعی و یادگیری ماشین

AI / Machine Learning Engineering



از داده تا مدل: ساختن سیستم‌هایی که یاد می‌گیرند

5/5

سختی یادگیری



5/5

برنامه‌نویسی جودسازی



4/5

امکان دورکاری



5/5

تقاضای بازار



مهندسی هوش مصنوعی، مدارهای یادگیری ماشین و یادگیری عمیق، با طراحی، آموزش و بهبود مستمر مدل‌ها، شبکه‌های عصبی، پردازش زبان طبیعی و ابزار هوش مصنوعی، با پایتون و فریمورک‌هایی مثل PyTorch، دارا. یادگیری حوزه فناوری‌هایی سایه‌ساز، اما همین حدی برای رقابت برای موقعیت‌های ارشد را هم بالا برده.

چرا این حوزه سودآور است

- سرمایه‌گذاری غول‌های فناوری روی هوش مصنوعی، بی‌سابقه است
- کاربردهای آن از پزشکی تا مالی، گسترده و پر به افزایش است
- نقش‌های تخصصی و سرعت پیشرفت، در شرکت‌های بین‌المللی وجود دارد

بازار کار جهانی

پدرآمدترین و پرتقاضاترین، حوزه، فناوری‌های، شغلی، بازنند، دانش‌ها شرکت‌های بزرگ دنیا.

≈ ۱۲۰,۰۰۰ تا ۲۴۰,۰۰۰ دلار سالانه یا بیشتر

بازار کار در ایران

رشد سریع اما رقابت شدید برای متخصصانی که مهارت بالا و تجربه کافی داشته باشند، فرصت‌های خوب است.

≈ ۳۰ تا ۷۰ میلیون تومان در ماه

بلاک‌چین و وب۳

Blockchain / Web3 Development



زنجیره‌ای فراتر از فردا! به دنیا غیرمتمرکز، وابسته نیست

4/5

سختی یادگیری



5/5

برنامه‌نویسی جودسازی



5/5

امکان دورکاری



3/5

تقاضای بازار



توسعه‌دهنده بلاک‌چین، قراردادهای هوشمند و اپلیکیشن‌های غیرمتمرکز را با زبان‌هایی مانند Solidity و Rust می‌نویسد. این حوزه نسبت به بقیه بازار نچیره‌ی و سرمایه است و همراه با رونق و رشد بازار رمزارزها، فرصت‌های شغلی هم رشد با ثبات می‌کنند.

چرا این حوزه سودآور است

- پروژه‌های وب۳ اغلب پودده بالا و پرداخت‌های سریع دارند.
- تعداد توسعه‌دهنده‌های متمرکز هوشوز کم و تقاضا برای آن‌ها بالاست.
- بیشتر پروژه‌ها به صورت کاملاً ریموت و بین‌المللی اجرا می‌شوند.

بازار کار جهانی

در دیفای، وب۳ و بازار رمزارز، از توسعه‌دهندگان به استفاده زیاد و بنیان‌گذاران، شرکت‌ها و تیم‌های بزرگ.

≈ ۱۰۰,۰۰۰ تا ۲۵۰,۰۰۰ دلار سالانه یا بیشتر

بازار کار در ایران

بازار محدودتر و به نسبت وابسته به ارتباط با پروژه‌های بین‌المللی. شرکت‌های نوپا و استارت‌آپ‌ها است.

≈ ۴۰ تا ۱۰۰ میلیون تومان در ماه (یسته به پروژه)

مهندسی قابلیت اطمینان سایت

Site Reliability Engineering (SRE)

تکنیک‌های سیستم‌های بزرگ، مقیاس‌پذیر، پسرار حساس



4/5

سنخ پذیر



4/5

برنامه‌نویسی توزیع‌پذیر



5/5

امکان دورکاری



4/5

تقاضای بازار



مهندسی SRE با استفاده از اصول مهندسی نرم‌افزار، بازیابی، مقیاس‌پذیری و مانیتورینگ سیستم‌های بزرگ را تضمین می‌کند. این نقش ترکیبی از برنامه‌نویسی و عملیات زیرساخت است و معمولاً در شرکت‌هایی با ترافیک بالا و حساسیت شدید به قطعی، به همراه ارائه‌دهنده خدمات می‌پردازد.

چرا این حوزه سودآور است

- قطعی سرویس برای شرکت‌های بزرگ هزینه مالی و اعتباری سنگینی دارد
- این شغل معمولاً بالاترین سطح دسترسی و مسئولیت همراه است
- به دلیل کمبود نیروی متخصص، حقوق آن اغلب از میانگین معمولی بالاتر است.

بازار کار جهانی

در شرکت‌های بزرگ فناوری، جهان یک شغلی با تقاضای بالا برای مهندسان اطلاعاتی پیوسته.

~۱۷۰,۰۰۰ تا ۳۵۰,۰۰۰ دلار سالانه

بازار کار در ایران

عمدتاً در شرکت‌های فناوری، بزرگ و شرکت‌های اینترنتی، استارت‌آپ‌ها و بخش‌های زیرساخت فعالیت می‌کنند.

~۲۵ تا ۴۰ میلیون تومان در ماه

بک‌اند و فول‌استک

Backend / Full-Stack Development

موتور پنهان هر محصول دیجیتال



3/5

سنخ پذیر



5/5

برنامه‌نویسی توزیع‌پذیر



5/5

امکان دورکاری



5/5

تقاضای بازار



توسعه‌دهنده بک‌اند و فول‌استک، مسئول سرور، پایگاه‌داده و API محصولات نرم‌افزاری یا وبسایت است. این حوزه پرتقاضاترین بازار کار فنی است، به همین دلیل رقابت برای موقعیت‌های خوب هم بیشتر از حوزه‌های تخصصی‌تر است.

چرا این حوزه سودآور است

- تقاضا هم محصول دیجیتال به یک بک‌اند نیاز دارد
- حجم بالای فرصت شغلی، ورود به بازار کار با درآمدهای متنوع
- شخصیت پر یک اکوسیستم خاص (مثل Node.js یا Go) می‌تواند حقوق را به سرعت بالا ببرد

بازار کار جهانی

بازار کار جهانی، پلنفرم‌های شرکت‌های بزرگ و استارت‌آپ‌ها به شدت به بک‌اند و فول‌استک‌ها نیاز دارند. این مشاغل معمولاً در رده درآمد بالا قرار دارند.

~۵۰,۰۰۰ تا ۸۵,۰۰۰ دلار سالانه

بازار کار در ایران

شرکت‌های بزرگ، کسب‌وکارهای اینترنتی، استارت‌آپ‌ها و شرکت‌های نرم‌افزاری به شدت به این تخصص نیاز دارند.

~۲۰ تا ۳۵ میلیون تومان در ماه

مقایسه پک‌نگاه هشت مسیر شغلی

پیش از انتخاب مسیر، این جدول را به عنوان نقشه راه اولیه در نظر بگیرید؛ اما مجدداً دانسته باشید که علاقه و پشتکار شخصی، مهم‌تر از هر عددی در این جدول است.

حوزه	سختی	برنامه‌نویسی	دورکاری	تقاضای بازار
● مهندسی نرم‌افزار	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● دواپس	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● مهندسی داده	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● امنیت تهاجمی و سایبری	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● هوش مصنوعی و یادگیری ماشین	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● بلاکچین و وب ۳	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● مهندسی قابلیت اطمینان سایت	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■
● پک‌اند و فرانت‌اند	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■	■ ■ ■ ■ ■ ■ ■ ■

یک نکته صادقانه

همه ارقام، فقط در این جدول صرفاً به عنوان یک دید کلی از وضعیت فعلی بازار در سال‌های ۲۰۲۵ تا ۲۰۲۶ هستند و یسته به شرکت، سطح تجربه، مهارت زبان انگلیسی و شرایط اقتصادی می‌توانند تفاوت محسوسی داشته باشند. بهترین مسیر، مسیریست که هم بازار برایش تقاضا دارد و هم شما برای یادگیری، ماندگاری، علاقه‌مند باشید؛ چون در نهایت، یادگیری، هرگز متوقف نمی‌شود.

بخش اول: تگ‌های پرکاربرد HTML

HTML زبان نشانه‌گذاری صفحات وب است و ساختار و محتوای صفحه را با «تگ» مشخص می‌کند. بیشتر تگ‌ها به صورت جفت (باز و بسته) نوشته می‌شوند، اما برخی مانند `img` و `br` تک‌بخشی‌اند. جدول‌های زیر مهم‌ترین تگ‌ها را با توضیح کوتاه و مثال نشان می‌دهند.

ساختار پایه سند

<code><!DOCTYPE html></code>	نوع سند را HTML5 اعلام می‌کند؛ اولین خط هر فایل.	<code><!DOCTYPE html></code>
<code><html lang="fa" dir="rtl"></code>	ریشه سند؛ همه تگ‌ها داخل آن‌اند. ویژگی <code>lang</code> و <code>dir</code> را اینجا بگذارید.	<code><html></code>
<code><head> ... </head></code>	اطلاعات نامرئی صفحه (عنوان، استایل، متا) در این بخش قرار می‌گیرد.	<code><head></code>
<code><title>My Page</title></code>	عنوان صفحه که در تب مرورگر و نتایج گوگل دیده می‌شود.	<code><title></code>
<code><meta charset="UTF-8"></code>	متادیتا: کدگذاری، توضیحات سئو، تنظیم نمایش موبایل.	<code><meta></code>
<code><link rel="stylesheet" href="style.css"></code>	اتصال فایل خارجی (معمولاً CSS) به صفحه.	<code><link></code>
<code><style> p{color:red} </style></code>	نوشتن CSS مستقیم داخل صفحه.	<code><style></code>
<code><script src="app.js"></script></code>	افزودن یا اتصال کد جاوااسکریپت.	<code><script></code>

متن و عنوان‌ها

<code><h1>Main title</h1></code>	عنوان‌ها؛ <code>h1</code> مهم‌ترین و <code>h6</code> کم‌اهمیت‌ترین. برای هر صفحه یک <code>h1</code> کافی است.	<code><h1> تا <h6></code>
<code><p>This is a paragraph.</p></code>	پاراگراف متن.	<code><p></code>
<code>line 1
line 2</code>	رفتن به خط بعد (تگ تک‌بخشی).	<code>
</code>
<code><hr></code>	یک خط افقی برای جداسازی بخش‌ها.	<code><hr></code>
<code>Important</code>	متن مهم؛ معمولاً پررنگ نمایش داده می‌شود.	<code></code>
<code>Note</code>	تأکید در متن؛ معمولاً کج نمایش داده می‌شود.	<code></code>
<code>text</code>	محفظه درون‌خطی برای استایل‌دهی بخشی از متن.	<code></code>
<code><blockquote>Quote</blockquote></code>	نقل قول بلند از یک منبع.	<code><blockquote></code>
<code><pre><code>x = 5</code></pre></code>	نمایش متن با حفظ فاصله‌ها و نمایش کد برنامه‌نویسی.	<code><pre></code> و <code><code></code>





بخش دوم: ویژگی‌های پرکاربرد CSS

CSS زبان استایل‌دهی است و ظاهر عناصر HTML را تعیین می‌کند. ساختار کلی هر قانون: انتخابگر { ویژگی: مقدار; }

روش‌های افزودن CSS

<code><p style="color:red;"></code>	داخل ویژگی style همان تگ (کمترین اولویت مناسب‌بودن، بیشترین قدرت).	Inline
<code><style> p{color:red;} </style></code>	داخل تگ style در head.	Internal
<code><link rel="stylesheet" href="a.css"></code>	فایل جدا و اتصال با link (بهترین روش).	External

رنگ و پس‌زمینه

<code>color: #333;</code>	رنگ متن.	color
<code>background-color: #f5f5f5;</code>	رنگ پس‌زمینه.	background-color
<code>background-image: url(bg.jpg);</code>	تصویر یا گرادیان پس‌زمینه.	background-image
<code>background-size: cover;</code>	اندازه تصویر پس‌زمینه: cover یا contain.	background-size
<code>background-repeat: no-repeat;</code>	تکرار تصویر پس‌زمینه.	background-repeat
<code>opacity: 0.8;</code>	شفافیت عنصر از 0 تا 1.	opacity
<code>background: linear-gradient(red, blue);</code>	پس‌زمینه گرادیان (رنگ‌های ترکیبی).	linear-gradient()





واحد‌های اندازه‌گیری

px	پیکسل؛ واحد ثابت.	font-size: 16px;
%	درصدی از اندازهٔ والد.	width: 50%;
em	نسبت به اندازهٔ فونت عنصر (یا والد).	padding: 1.5em;
rem	نسبت به اندازهٔ فونت ریشه (html)؛ مناسب برای ریسپانسیو.	font-size: 1.2rem;
vw / vh	درصدی از عرض / ارتفاع پنجرهٔ مرورگر.	height: 100vh;
fr	سهمی از فضای خالی در Grid.	grid-template-columns: 1fr 2fr;

متن و فونت

font-family	نوع فونت (با فونت جایگزین).	font-family: Tahoma, sans-serif;
font-size	اندازهٔ متن (px, em, rem).	font-size: 16px;
font-weight	ضخامت متن: normal, bold, 100 تا 900.	font-weight: bold;
font-style	حالت متن؛ italic برای کج‌نویسی.	font-style: italic;
text-align	تراز افقی متن: left, right, center, justify.	text-align: center;
line-height	ارتفاع خط؛ برای خوانایی مهم است.	line-height: 1.8;
text-decoration	زیرخط و خط‌خورده؛ برای حذف زیرخط لینک .none	text-decoration: none;
text-transform	تبدیل حروف: uppercase, lowercase, capitalize.	text-transform: uppercase;
letter-spacing	فاصلهٔ بین حروف.	letter-spacing: 1px;

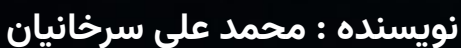


مدل جعبه (Box Model)

width: 300px;	عرض و ارتفاع محتوای عنصر.	width / height
max-width: 1200px;	حداکثر و حداقل عرض؛ مهم در طراحی واکنش‌گرا.	max-width / min-width
padding: 10px 20px;	فاصله داخلی بین محتوا و حاشیه.	padding
margin: 0 auto;	فاصله بیرونی با عناصر اطراف؛ 0 برای وسط‌چین بلوک.	margin
border: 1px solid #ccc;	حاشیه: ضخامت، نوع، رنگ.	border
border-radius: 8px;	گرد کردن گوشه‌ها؛ 50% برای دایره.	border-radius
box-shadow: 0 2px 6px rgba(0,0,0,.2);	سایه دور عنصر.	box-shadow
box-sizing: border-box;	محاسبه اندازه؛ border-box padding و border را داخل عرض حساب می‌کند.	box-sizing
overflow: hidden;	رفتار محتوای سرریز: hidden, auto, scroll.	overflow

نمایش و موقعیت‌دهی

display: none;	نوع نمایش: block, inline, inline-block, none, flex, grid.	display
position: relative;	نوع موقعیت: static, relative, absolute, fixed, sticky.	position
top: 0; right: 0;	جابه‌جایی عنصر موقعیت‌دار.	top / right / bottom / left
z-index: 10;	ترتیب لایه‌ها؛ عدد بزرگ‌تر بالاتر است.	z-index
cursor: pointer;	شکل نشانگر ماوس.	cursor



BOOT.DEV

وقتی یادگیری برنامه‌نویسی حس و حال یک بازی نقش‌آفرینی پیدا می‌کند



با نوشتن یک عالمه کد
خودتو سرگرم کن



افزایش انگیزه
یادگیری با بازی



ساخت پروژه‌های نمونه‌کار
نمونه‌کار برای اثبات مهارت



با هزینه کمتر از
آموزشگاه‌ها



به صورت آنلاین و
منعطف



ورود عمیق به مفاهیم پایه

آیا می‌دانستید که یادگیری می‌تواند به اندازه بازی‌های نقش‌آفرینی مورد علاقه‌تان سرگرم‌کننده باشد؟

نخستین تجربه من از کار با Boot.dev، مرا به روزهایی برد که ساعت‌ها غرق در بازی Diablo III بودم. این حس نوستالژیک از فضا سازی هدفمند پلتفرم نشئت می‌گیرد؛ از کوئست‌ها (که در واقع همان درس‌ها هستند) گرفته تا سازوکارهای بازی گونه‌ای که شما را به این توهم می‌اندازد که درون یک دنیای مجازی قرار گرفته‌اید.

صادقانه بگویم، در ابتدا چندان تحت تأثیر قرار نگرفته بودم؛ اما پس از اتمام کل مسیر توسعه بک‌اند (Backend Development)، با قاطعیت می‌توانم بگویم که این پلتفرم به بیشتر وعده‌های خود وفا کرده است. تیم سازنده Boot.dev محیطی سرگرم‌کننده برای یادگیری ایجاد کرده که نه تنها فرایند آموزش را لذت‌بخش، بلکه کارآمد نیز می‌سازد. تمام جزئیات با هدفی مشخص طراحی شده‌اند و تجربه یادگیری برنامه نویسی را به ماجراجویی‌ای برای دستیابی به سطحی بالاتر تبدیل می‌کنند.

آیا این تجربه به اندازه سرمایه گذاری زمان و انرژی‌اش ارزشمند بود؟

پاسخ به این سؤال قاطعانه مثبت است. پیش از شروع این مسیر، با مبانی پایتون و توسعه وب آشنایی داشتم؛ اما در مورد امکان ادامه هم‌زمان آن با تحصیل دانشگاهی و برنامه روزانه‌ای که داشتم، تردیدهایی وجود داشت. با این وجود، به طور منظم هر روز باز می‌گشتم و در بیشتر روزها حداقل یک ساعت مطالعه داشتم. از نظر انگیزه نیز هیچ مشکلی احساس نکردم.

در ۲۸ سالگی و با یک برنامه روزانه شلوغ، انتظار نداشتم که صرفاً با اتمام این دوره‌ها، یک باره به عنوان توسعه‌دهنده‌ای حرفه‌ای مطرح شوم. پلتفرم به وضوح تأکید دارد که باید نمونه‌کار (Portfolio) تهیه کنید، با دیگران ارتباط برقرار کنید و از دوره نهایی نیز برای آشنایی با فرایند کاریابی استفاده کنید؛ دوره‌ای که من هنوز آن را تکمیل نکرده‌ام. احساس نمی‌کنم کاملاً برای ورود به بازار کار آماده باشم، اما این موضوع بیشتر به ویژگی‌های شخصی من مربوط می‌شود. از نظر دانش و مهارت، به نظر می‌رسد برای پذیرش یک موقعیت شغلی در سطح ابتدایی به عنوان توسعه‌دهنده بک‌اند آمادگی کافی داشته باشم. به بیان دیگر، می‌توانم بسیاری از کارها را به تنهایی انجام دهم، اما تا زمانی که کاملاً مستقل نشده‌ام، به راهنمایی یک متخصص باتجربه نیاز دارم. با این حال، باید انصاف را رعایت کرد Boot.dev: مهارت‌های عملی و ضروری‌ای را به من یاد داد که برای شروع کار به عنوان یک توسعه‌دهنده به آن‌ها نیاز داشتم؛ از نوشتن و دیباگ کردن کد گرفته تا مطالعه و استفاده از مستندات، کار با Git و GitHub، Docker و ابزارهای مختلف توسعه.



ساختار مسیر آموزشی؛ از کجا شروع می‌شود و چگونه پیش می‌رود؟

مسیر یادگیری به‌گونه‌ای طراحی شده است که شما را قدم‌به‌قدم به یک توسعه دهندهٔ بک اند تبدیل کند. این مسیر با آموزش پایتون آغاز می‌شود و سپس به سراغ مباحث کلیدی دیگری می‌رود: کار با شل و ترمینال، سیستم کنترل نسخهٔ Git، پروژه‌های عملی برای تثبیت یادگیری، الگوریتم‌ها و ساختارهای داده و در نهایت زبان‌های برنامه‌نویسی Go و JavaScript. در مجموع، این مسیر اصلی از 24 دوره تشکیل شده است که هرکدام شما را برای مرحلهٔ بعدی آماده می‌کنند.

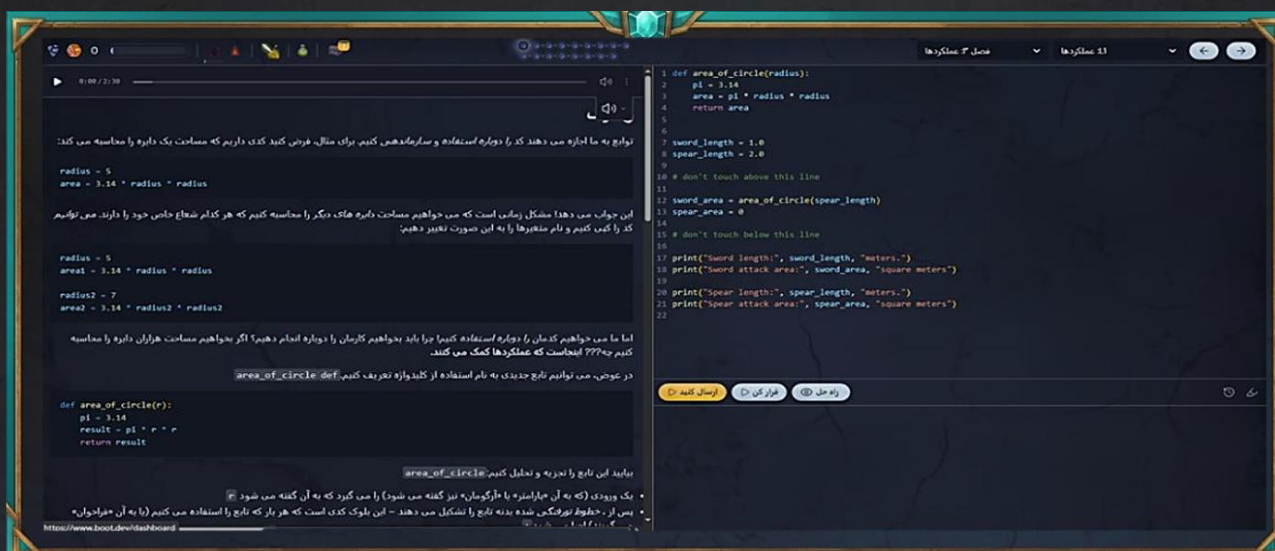
دورهٔ پایتون، با بیش از ۱۶۰ درس، به مراتب جامع‌ترین بخش مسیر یادگیری است. من خودم این دوره را در حدود ۲۰ ساعت به پایان رساندم. این حجم بالای محتوا کاملاً قابل درک است؛ زیرا در این دوره با اصول بنیادینی آشنا می‌شوید که نه تنها در پایتون، بلکه در هر زبان برنامه‌نویسی دیگری نیز کاربرد دارند: از متغیرها و توابع گرفته تا تست نویسی، اشکال زدایی (Debugging) و مبانی معماری نرم‌افزار. به عبارت دیگر، این دوره زیرساخت ذهنی لازم برای یادگیری هر زبان دیگری را در شما ایجاد می‌کند. خوشبختانه، پس از اتمام موفقیت آمیز این دوره، یک گواهی رسمی (Certificate) نیز دریافت می‌کنید که می‌تواند به رزومهٔ شما اعتبار ببخشد.

برای روشن تر شدن میزان پیشرفت، باید بگویم که موفق شدم 16 دوره و 8 پروژهٔ راهنما را با موفقیت به پایان برسانم. در بیشتر روزها حداقل یک ساعت مطالعهٔ منظم داشتم و این روند را چنان ادامه دادم که به سطح ۳۰ دست یافتم. نکتهٔ جالب‌تر اینکه طبق قوانین پلتفرم، کاربرانی که به سطح ۱۰۰ می‌رسند، یک نامهٔ شخصی و یک سکه از طرف Boot.dev دریافت می‌کنند. این یادگاری فیزیکی، اگرچه کوچک است، ارزش معنایی بالایی دارد و نشان‌دهندهٔ تداوم تلاش‌های انجام شده است.

جایی که یادگیری به بازی تبدیل می‌شود: نقش بازی‌وارسازی

یکی از جذاب‌ترین جنبه‌های Boot.dev، غرق شدن در فضای بازی گونهٔ آن است. آن‌قدر ویژگی‌های سرگرم کننده در این پلتفرم تعبیه شده که به راحتی می‌توان فراموش کرد در حال یادگیری هستید، نه بازی کردن. نکتهٔ جالب‌تر، ارجاع‌های بامزه‌ای است که در سراسر پلتفرم پراکنده شده‌اند؛ مثل استفاده از نام شخصیت‌های معروف ارباب حلقه‌ها به‌عنوان آرگومان در توابع برنامه‌نویسی. این جزئیات کوچک اما هوشمندانه، حس نزدیکی و صمیمیت را برای کاربرانی که با این فرهنگ آشنا هستند، دوچندان می‌کنند.





سیستم پاداش دهی در Boot.dev به گونه ای طراحی شده است که هر بار ۱۵ تمرین متوالی را بدون خطا پشت سر بگذارید، یک صندوقچه تصادفی (Random Chest) دریافت می کنید که شامل ترکیبی از جواهرات (Gems) و آیتم های متنوع است. این آیتم ها را می توان به عنوان تقویت کننده در مسیر یادگیری به کار برد. برای نمونه، آیتم «ماهی سالمون پخته (Baked Salmon)» به شما امکان می دهد بدون کسر امتیاز تجربه (XP) از کمک های Boots، دستیار هوش مصنوعی، استفاده کنید. همچنین «سنگ غیب گو (Seer Stones)» این امکان را فراهم می کند که پاسخ تمرین ها را بدون هیچ جریمه ای مشاهده کنید. این آیتم ها را می توان با جواهرات به دست آمده از صندوقچه ها خریداری کرد.



سیستم دستاوردها (Achievements) دارای ۷ دسته بندی است:

- نقش (Role): وابسته به سطح شما
- تداوم (Streak): تعداد روزهای مطالعه پیوسته
- نقطه عطف (Milestone): تعداد تمرین های تکمیل شده
- تیرانداز دقیق (Sharpshooter): تعداد زنجیره های بدون اشتباه
- سالار پرچم (Bannerlord): مأموریت دعوت از یک دوست
- هم پیمانی (Fellowship): کسب ۱ کارما در دیسکورد

Boots؛ همراه همیشگی هوش مصنوعی شما در مسیر یادگیری

یکی از قابلیت‌های شاخص Boot.dev، چت بات هوشمند آن یعنی Boots است. من معمولاً در هر فصل یک یا دو بار سراغ Boots می‌روم؛ گاهی برای اشکال زدایی کد و گاهی پس از اتمام درس، وقتی هنوز درباره موضوعی تردید دارم.

یک میانبر بسیار کاربردی وجود دارد که با یک کلیک، کد شما را مستقیماً در چت قرار می‌دهد. تجربه کار با Boots دقیقاً شبیه صحبت با ChatGPT یا Claude است، با این تفاوت که Boots دقیقاً می‌داند در کدام مرحله از درس هستید و زمینه کامل تری از محتوای آموزشی در اختیار دارد. جالب‌ترین ویژگی آن این است که از روش سقراطی استفاده می‌کند؛ یعنی به جای اینکه پاسخ را مستقیماً به شما بدهد، با پرسش‌های هدفمند شما را به سمت کشف راه حل هدایت می‌کند.

علاوه بر Boots، یک منبع کمک آموزشی دیگر نیز وجود دارد که کاملاً رایگان و بدون هیچ جریمه‌ای در دسترس است: کتاب طلسم‌ها (Spellbook). این بخش در واقع یک دانشنامه شخصی است که تمام درس‌هایی را که تاکنون گذرانده‌اید، به صورت خلاصه و در دسترس نگه می‌دارد تا هر زمان نیاز داشتید، آن‌ها را مرور کنید.

محوریت اصلی Boot.dev بر جامعه کاربران آن است. در نخستین درس‌ها، چندین بار برای پیوستن به سرور دیسکورد دعوت می‌شوید و پس از ورود، اهمیت آن را به خوبی درک می‌کنید. اینجا جامعه‌ای پرجنب و جوش از زبان آموزان و یادگیرندگان شکل گرفته است.

دیسکورد برای دریافت اعلان‌ها و اطلاعیه‌های رسمی، گفت‌وگو با سایر کاربران، دریافت مشاوره درباره صاحب‌های شغلی، یافتن فرصت‌های شغلی و آشنایی با مسیرهای حرفه‌ای و موارد مشابه کاربرد دارد.

باید اعتراف کنم که در طول دوره یادگیری، به اندازه دلخواه فرصت تعامل در دیسکورد را نداشتم. با این حال، کاربران و مدیران (Moderators) بسیار فعال و پشتیبان هستند. من هر از گاهی به چند سؤال پاسخ داده‌ام و کاش بیشتر در آنجا حضور داشتم که همین موضوع باعث شد احساس کنم بخشی از یک جمع بزرگ هستم.

حرف آخر؛ آیا این مسیر برای شما مناسب است؟

در آغاز این مسیر، کنج‌کاو بودم که آیا این پلتفرم واقعاً قادر به آموزش توسعه بک اند (Backend) است یا خیر. اکنون با اطمینان می‌توانم بگویم که چنین است، البته به شرط تداوم و پابندی به مطالعه منظم.

Boot.dev به تعهد خود عمل کرد: مرا درگیر نگه داشت، مبانی اساسی را آموزش داد و ابزارهای لازم برای ادامه‌ی مسیر را در اختیارم قرار داد. نکات بسیاری درباره‌ی سازوکارهای پلتفرم و ویژگی‌های دقیقاً طراحی‌شده‌ی آن وجود دارد که جای بحث دارد، اما ترجیح می‌دهم بخشی از آن را برای شما به عنوان یک کشف شخصی باقی بگذارم.

من خود را یک توسعه‌دهنده ارشد (Senior) نمی‌دانم و چنین ادعایی هم ندارم. اما اکنون توانایی ساخت پروژه‌های عملی را دارم و این دقیقاً همان هدفی بود که در ابتدا دنبال می‌کردم. پلتفرم به خوبی ساخته شده، محتوای دوره‌ها از کیفیت نگارش و طراحی مناسبی برخوردار است، تمرین‌ها به اندازه کافی چالش برانگیز هستند و منابع کمکی متعددی برای مواقع نیاز در دسترس است.

اگر مایل به سرمایه گذاری زمانی باشید، Boot.dev در نیمه راه با شما همراه خواهد شد. دیگر بهانه‌ای برای توقف یادگیری هنگام مواجهه با مشکل وجود ندارد و این تجربه‌ای لذت بخش است که شما را متعهد نگه می‌دارد و بارها برای بازگشت ترغیب می‌کند.

وقت اضافه



اولین هارددیسک
تجاری IBM حدود
۵ مگابایت ظرفیت
داشت ، اما تقریباً
به اندازه یک اتاق
بود !



اولین ماوس کامپیوتر
در دهه ۱۹۶۰ ساخته
شد و بدنه ای چوبی
داشت .



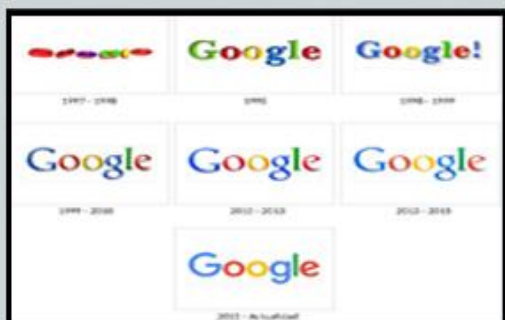
+5:00



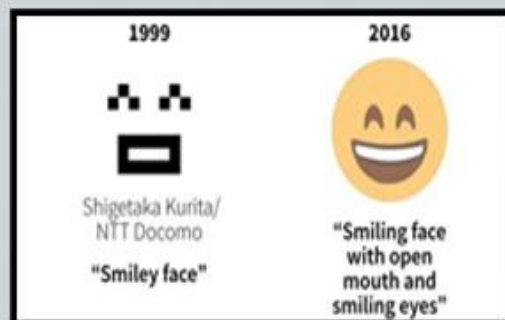
یکی از مشهورترین داستان های تاریخ کامپیوتر مربوط به پیدا شدن یک حشره واقعی داخل دستگاهی در سال ۱۹۴۷ است .



اولین وبسایت جهان در سال ۱۹۹۱ راه اندازی شد ؛ زمانی که ظاهر وب تقریباً هیچ شباهتی به اینترنت امروز نداشت .



نام Google به واژه ی Googol ، یعنی عدد ۱ با ۱۰۰ صفر ، مرتبط است .



مجموعه معروف ۱۷۶ تایی ایموجی در سال ۱۹۹۹ در ژاپن طراحی شد .



+5:00

دور نگاه

رایانه‌های کوانتومی؛ نسل آینده‌ی پردازش اطلاعات

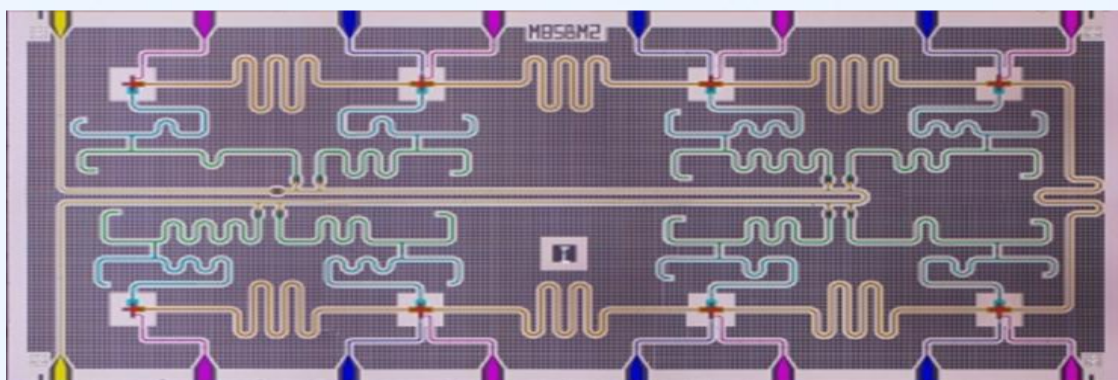
محاسبات کوانتومی بی‌شک یکی از گسترده‌ترین و در عین حال چالش‌برانگیزترین حوزه‌ها در میان فناوری‌های کوانتومی به‌شمار می‌رود. آنچه یک رایانه‌ی کوانتومی را از رایانه‌های کلاسیک متمایز می‌سازد، دو ویژگی بنیادین مکانیک کوانتومی است: نخست، کیوبیت‌ها (qubits) که برخلاف بیت‌های کلاسیک، می‌توانند هم‌زمان در حالت‌های صفر و یک قرار گیرند (برهم‌نهی یا superposition)، و دوم، همبستگی‌های آنی بین کیوبیت‌ها در سراسر دستگاه (درهم تنیدگی یا entanglement). این ویژگی‌ها به رایانه‌ی کوانتومی امکان می‌دهد تا به‌عنوان یک ماشین موازی عظیم عمل کند که تعداد نمایی از محاسبات را به‌طور هم‌زمان انجام می‌دهد. امروزه الگوریتم‌های متعددی طراحی شده‌اند که از این قدرت بی‌نظیر بهره می‌برند و راه را برای حل مسائلی هموار می‌کنند که حتی قوی‌ترین ابررایانه‌های کلاسیک نیز هرگز توانایی حل آن‌ها را نخواستند.

نویسنده : مهران جاوید



در دو دهه‌ی گذشته، نمونه‌های متعددی از رایانه‌های کوانتومی با پلتفرم‌های مختلف به جهان معرفی شده‌اند. در میان این رویکردها، دو فناوری بیشتر از همه پیشرفت کرده‌اند: یون‌های به دام افتاده (trapped ions) و مدارهای ابررسانا (superconducting) circuits. نمونه‌های اولیه‌ی این رایانه‌ها با ظرفیت محدود ۱۰ تا ۱۵ کیوبیت، تاکنون توانسته‌اند الگوریتم‌ها و پروتکل‌های ساده‌ی کوانتومی را با موفقیت اجرا کنند و گام‌های اولیه را در مسیر تحقق رایانش کوانتومی عملی بردارند.

در سال‌های اخیر، پلتفرم‌ها و معماری‌های متعددی موفق به نمایش اصول اولیه‌ی رایانش کوانتومی شده‌اند. این تنوع گسترده، هم شامل سیستم‌های حالت جامد (مانند اسپین‌های الکترونی در نیمه‌رساناها، اسپین‌های هسته‌ای در جامدات و حالت‌های صفر ماژورانا) می‌شود و هم سیستم‌های اتمی و نوری (از جمله اسپین‌های هسته‌ای در مولکول‌ها، حالت‌های فوق‌ریز و ریدبرگ در اتم‌ها و فوتون‌ها) را در بر می‌گیرد. این تنوع نشان‌دهنده‌ی غنای این حوزه و رویکردهای متعدد برای دستیابی به رایانش کوانتومی عملی است.



پردازنده‌ی کوانتومی ابررسانای ۸-کیوبیتی ساخته‌شده در دانشگاه صنعتی زوریخ (ETH Zurich)

بر اساس نقشه راه پیش‌بینی‌شده برای توسعه‌ی رایانش کوانتومی، سه نشان‌های کلیدی در افق‌های زمانی مختلف ترسیم شده‌اند:

- **۳ سال آینده:** مسیرهای مقاوم در برابر خطا (fault-tolerant) برای ساخت پردازنده‌های کوانتومی با ظرفیت نهایی بیش از ۵۰ کیوبیت، به صورت عملی به نمایش درخواهند آمد و امکان‌پذیری فنی این رویکرد تأیید خواهد شد.
- **۶ سال آینده:** پردازنده‌های کوانتومی مجهز به تصحیح خطای کوانتومی (quantum error correction) یا کیوبیت‌های مقاوم (robust qubits) تحقق خواهند یافت. این پردازنده‌ها عملکردی برتر از کیوبیت‌های فیزیکی معمولی خواهند داشت و گامی مهم به سوی رایانش کوانتومی مقیاس‌پذیر برداشته خواهد شد.

- **۱۰ سال آینده:** الگوریتم‌های کوانتومی که افزایش سرعت کوانتومی (quantum speed-up) را به وضوح نشان می‌دهند و از رایانه‌های کلاسیک در حل مسائل خاص پیشی می‌گیرند، به صورت عملیاتی و در مقیاس واقعی اجرا خواهند شد.

شبیه‌سازهای کوانتومی؛ پنجره‌ای نو به سوی شبیه‌سازی مواد و واکنش‌های شیمیایی

امروزه طراحی اشیای پیچیده‌ای مانند هواپیماها، ساختمان‌های بلند، خودروها و بسیاری از محصولات صنعتی دیگر، بدون استفاده از ابررایانه‌های قدرتمند امکان‌پذیر نیست. با این حال، ابررایانه‌های کلاسیک در برخی حوزه‌ها با محدودیت‌های جدی مواجه هستند. برای نمونه، ما هنوز نمی‌توانیم به دقت پیش‌بینی کنیم که یک ماده‌ی متشکل از تنها چند صد اتم، رسانای الکتریسیته خواهد بود یا خاصیت مغناطیسی از خود نشان می‌دهد؛ یا اینکه یک واکنش شیمیایی خاص تحت شرایط معین رخ خواهد داد یا خیر.

نقشه‌ی راه توسعه‌ی شبیه‌سازهای کوانتومی؛ اهداف کلیدی در افق‌های زمانی مختلف

بر اساس برنامه‌های پیش‌بینی‌شده برای توسعه‌ی شبیه‌سازهای کوانتومی، سه هدف کلیدی در بازه‌های زمانی متفاوت ترسیم شده‌اند:

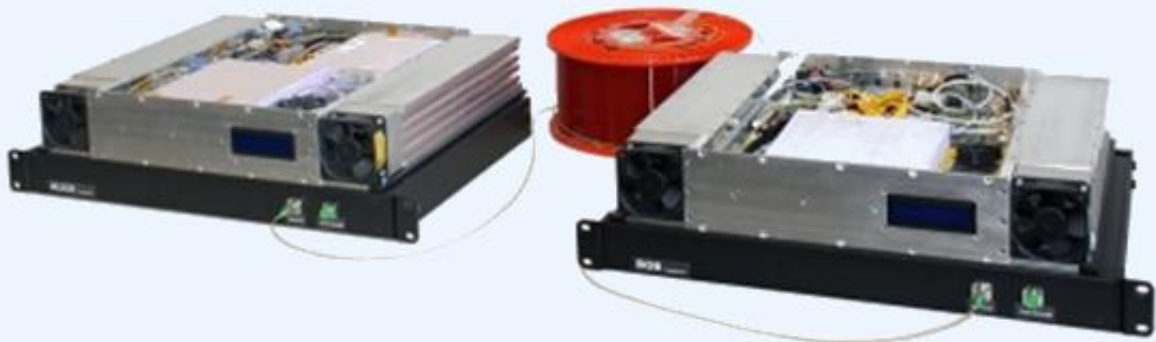
- **افق ۳ ساله:** انتظار می‌رود دستگاه‌های آزمایشی با قابلیت برتری کوانتومی تأییدشده (certified quantum advantage) محقق شوند؛ به‌گونه‌ای که بتوانند دست‌کم ۵۰ سیستم کوانتومی جفت‌شده‌ی منفرد (در معماری‌های پردازنده‌ای) یا ۵۰۰ سیستم (در شبکه‌های نوری) را در مقیاسی عملی به‌کار گیرند.

- **افق ۶ ساله:** در این مرحله، شبیه‌سازهای کوانتومی باید بتوانند برتری خود را در حل مسائل علمی بنیادین – مانند شبیه‌سازی پدیده‌های مغناطیس کوانتومی – به‌اثبات برسانند. همچنین نمایش عملی بهینه‌سازی کوانتومی، مثلاً از طریق روش بازپخت کوانتومی (quantum annealing)، به‌عنوان یکی دیگر از دستاوردهای کلیدی این دوره پیش‌بینی می‌شود.

- **افق ۱۰ ساله:** در این بازه، نمونه‌های اولیه‌ی شبیه‌سازهای کوانتومی باید توانایی حل مسائلی را داشته باشند که برای ابررایانه‌های کلاسیک غیرقابل حل هستند. این مسائل حوزه‌های گسترده‌ای را در بر می‌گیرند: از شیمی کوانتومی و طراحی مواد جدید گرفته تا مسائل پیچیده‌ی بهینه‌سازی که در قلب توسعه‌ی هوش مصنوعی قرار دارند.

ارتباطات کوانتومی؛ انتقال امن اطلاعات در عصر کوانتوم

امنیت ارتباطات برای تمامی ذی‌نفعان - از کاربران عادی گرفته تا شرکت‌های بزرگ و نهادهای دولتی - از اهمیت استراتژیک برخوردار است. در حال حاضر، ستون فقرات این امنیت بر پایه‌ی الگوریتم‌های رمزنگاری کلاسیک استوار است. اما با ظهور رایانه‌های کوانتومی، این الگوریتم‌ها آسیب پذیر خواهند شد؛ چراکه رایانه‌های کوانتومی توانایی شکستن بسیاری از روش‌های رمزنگاری فعلی را دارند. به‌همین دلیل، حوزه‌ی جدیدی به نام «رمزنگاری مقاوم در برابر کوانتوم (quantum-safe cryptography)» شکل گرفته است که هدف آن توسعه‌ی الگوریتم‌هایی است که حتی در برابر قدرتمندترین رایانه‌های کوانتومی نیز مقاوم باشند. خوشبختانه، راه‌حل‌های امن مبتنی بر رمزنگاری کوانتومی نیز هم‌اکنون به‌صورت تجاری در دسترس هستند و می‌توانند امنیتی کامل در برابر حملات کوانتومی فراهم آورند. علاوه بر این، تولید اعداد تصادفی کوانتومی (quantum random number generation) - یکی از ارکان اصلی و حیاتی در اکثر پروتکل‌های رمزنگاری به‌شمار می‌رود - نیز امروزه به‌عنوان یک فناوری تجاری قابل‌دسترس، نقش کلیدی در تضمین امنیت ارتباطات آینده ایفا می‌کند.



یک سیستم نمونه‌ی اولیه‌ی توزیع کلید کوانتومی (QKD) که در یکی از دانشگاه‌ها ساخته شده است، توانایی عملکرد خودکار (بدون نیاز به مداخله‌ی انسانی) و توزیع کلیدهای رمزنگاری را در فواصل بیش از ۳۰۰ کیلومتر دارد.

یک سیستم نمونه‌ی اولیه‌ی توزیع کلید کوانتومی (QKD) که در یکی از دانشگاه‌ها ساخته شده است، توانایی عملکرد خودکار (بدون نیاز به مداخله‌ی انسانی) و توزیع کلیدهای رمزنگاری را در فواصل بیش از ۳۰۰ کیلومتر دارد.

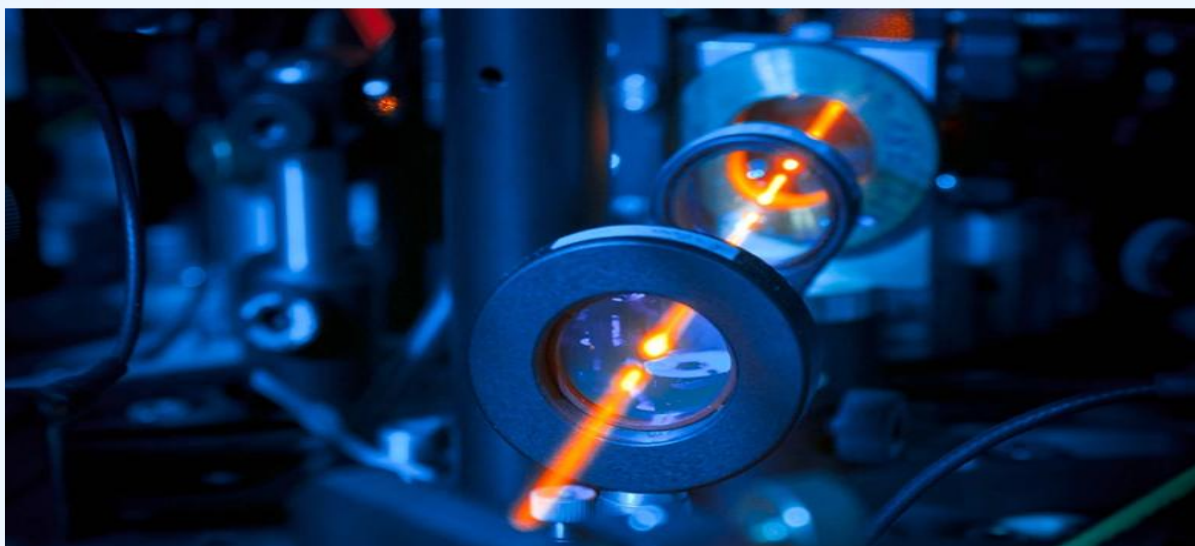
ارتباطات کوانتومی در مقیاس جهانی نیازمند زیرساخت‌های پیشرفته‌ای است که احتمالاً شامل ماهواره‌ها نیز خواهد بود. در این میان، تکرارکننده‌های کوانتومی (quantum repeaters) نقشی کلیدی ایفا می‌کنند، زیرا فاصله‌ی قابل‌پیمایش بین گره‌های قابل‌اعتماد را به‌طور قابل‌توجهی افزایش می‌دهند.

برای تحقق این تکرارکننده‌ها، دو بلوک سازنده‌ی اساسی مورد نیاز است: یک پردازنده‌ی کوانتومی کوچک و یک رابط کوانتومی که بتواند اطلاعات را به فوتون‌ها تبدیل کند. این رابط، عملکردی مشابه دستگاه‌های اپتوالکترونیکی موجود در اینترنت امروزی دارد، با این تفاوت که قابلیت‌های کوانتومی به آن افزوده شده است.

نقشه‌ی راه ارتباطات کوانتومی؛ اهداف کلیدی در افق‌های زمانی مختلف

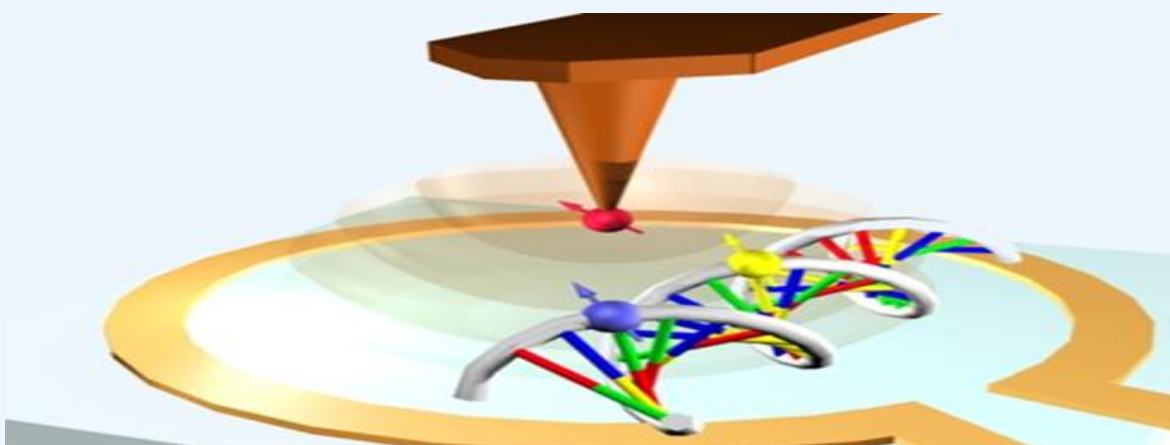
بر اساس برنامه‌های پیش‌بینی‌شده برای توسعه‌ی ارتباطات کوانتومی، سه افق زمانی کلیدی برای دستیابی به قابلیت‌های مختلف ترسیم شده است:

- **افق ۳ ساله:** تمرکز بر توسعه و تأییدیه‌ی رسمی (certification) دستگاه‌های تولید اعداد تصادفی کوانتومی (QRNG) و توزیع کلید کوانتومی (QKD). این دستگاه‌ها باید دارای سرعت بالا، سطح آمادگی فناوری پیشرفته (high-TRL)، هزینه‌ی پیاده‌سازی پایین و پشتیبانی از پروتکل‌های نوین باشند. هم‌زمان، پژوهش بر روی سیستم‌ها و پروتکل‌های مربوط به تکرارکننده‌های کوانتومی، حافظه‌های کوانتومی و ارتباطات دوربرد نیز آغاز خواهد شد.
- **افق ۶ ساله:** در این مرحله، دستگاه‌ها و سیستم‌هایی مقرون به صرفه و مقیاس پذیر برای شبکه‌های درون‌شهری و بین‌شهری ارائه می‌شوند که کاربردهای واقعی و الهام گرفته از نیاز کاربران نهایی را به‌نمایش می‌گذارند. همچنین راه‌حل‌های مقیاس‌پذیر برای شبکه‌های کوانتومی که بتوانند دستگاه‌های متنوعی مانند حسگرهای کوانتومی یا پردازنده‌های کوانتومی را به‌هم متصل کنند، به‌صورت عملی به‌نمایش درخواهند آمد.
- **افق ۱۰ ساله:** در بلندمدت، هدف توسعه‌ی شبکه‌های خودکار در مقیاس شهری، شبکه‌های دوربرد با برد بیش از ۱۰۰۰ کیلومتر، و شبکه‌های مبتنی بر درهم‌تنیدگی کوانتومی است که در مجموع به‌عنوان «اینترنت کوانتومی» شناخته می‌شوند. در این مرحله، پروتکل‌هایی نیز توسعه خواهند یافت که از ویژگی‌های منحصربه‌فرد ارتباطات کوانتومی – مانند امنیت مبتنی بر قوانین فیزیک و قابلیت‌های توزیع درهم‌تنیدگی – بهره‌برداری می‌کنند.



حسگرهای کوانتومی و مترولوژی؛ دقتی فراتر از محدودیت‌های کلاسیک

یکی از کاربردهای مهم فناوری کوانتومی در حوزه‌ی حسگری و اندازه‌گیری (مترولوژی) است. حالت‌های برهم نهی (superposition) در مکانیک کوانتومی به‌طور ذاتی به تغییرات محیطی بسیار حساس هستند و این ویژگی، آن‌ها را به ابزاری ایده‌آل برای ساخت حسگرهایی با دقتی بی‌سابقه تبدیل کرده است. طی سال‌های اخیر، پیشرفت‌های چشمگیری در کیفیت مواد، کنترل فرایندهای ساخت، کاهش هزینه‌ها و کوچک سازی اجزای کلیدی مانند لیزرها حاصل شده است. در نتیجه، این حسگرها اکنون از مرحله‌ی آزمایشگاهی خارج شده و آماده‌ی ورود به بازار و کاربردهای تجاری گسترده هستند. نمونه‌ای از این فناوری، حسگرهای کوانتومی حالت جامد هستند که بر پایه‌ی مراکز NV در الماس ساخته شده‌اند. این حسگرها توانایی اندازه‌گیری میدان‌های مغناطیسی بسیار ضعیف را دارند، به‌گونه‌ای که با روش‌های کلاسیک قابل دستیابی نیست و کاربردهای گسترده‌ای در حوزه‌های مختلف از علوم اعصاب تا اکتشافات معدنی خواهند داشت.



این تصویر، بازنمایی هنری از یک حسگر کوانتومی مبتنی بر اسپین را نشان می‌دهد که برای آشکارسازی و مطالعه‌ی ساختار تک‌مولکول‌های زیستی به‌کار می‌رود.

این موضوع به نوبه‌ی خود می‌تواند در کاربردهای متعددی مفید واقع شود

در حوزه‌ی تصویربرداری، فناوری‌های کوانتومی با استفاده از نور درهم‌تنیده (entangled light) انقلابی در استخراج اطلاعات از نور ایجاد کرده‌اند. این رویکرد امکانات جدیدی را فراهم می‌آورد: برای نمونه، استفاده از نور فشرده‌شده (squeezed light) می‌تواند وضوح تصاویر را به‌طور قابل‌توجهی افزایش دهد. همچنین امکان تولید تصویر با اندازه‌گیری تنها یک فوتون منفرد وجود دارد، به شرطی که این فوتون با فوتون دومی (با رنگی متفاوت) درهم‌تنیده باشد و آن فوتون دوم برای کاوش نمونه موردنظر به کار رود - تکنیکی که می‌تواند در حوزه‌های حساس به نور مانند زیست‌شناسی و پزشکی تحولی عظیم ایجاد کند. در حوزه‌ی ناوبری و اندازه‌گیری، دستگاه‌های تداخل‌سنج اتمی و مولکولی با تکیه بر پدیده‌ی برهم‌نهی کوانتومی، شتاب و چرخش را با دقتی بی‌سابقه اندازه‌گیری می‌کنند. این اندازه‌گیری‌ها می‌توانند در سیستم‌های ناوبری اینرسی (inertial navigation) به کار روند

نقشه‌ی راه حسگرها و مترولوژی کوانتومی؛ اهداف کلیدی در افق‌های زمانی مختلف

بر اساس برنامه‌های پیش‌بینی‌شده برای توسعه‌ی حسگرها و سیستم‌های اندازه‌گیری کوانتومی، سه افق زمانی کلیدی ترسیم شده‌است:

- **افق ۳ ساله:** در این بازه، انتظار می‌رود حسگرهای کوانتومی، سیستم‌های تصویربرداری و استانداردهای مترولوژی که از همدوسی تک‌کیوبیتی (single qubit coherence) بهره می‌برند، در محیط آزمایشگاهی به‌نمایش درآیند. این دستگاه‌ها باید در معیارهایی مانند وضوح تصویر و پایداری، عملکردی بهتر از نمونه‌های کلاسیک خود ارائه دهند.
- **افق ۶ ساله:** در این مرحله، نمونه‌های اولیه (prototype) از حسگرهای کوانتومی یکپارچه، سیستم‌های تصویربرداری و استانداردهای مترولوژی ساخته خواهند شد و نخستین محصولات تجاری وارد بازار می‌شوند. به موازات آن، فناوری‌های حسگری مبتنی بر درهم‌تنیدگی (entanglement-enhanced) نیز در سطح آزمایشگاهی به‌نمایش درخواهند آمد که نویدبخش دقتی فراتر از محدودیت‌های کلاسیک هستند.
- **افق ۱۰ ساله:** در بلندمدت، هدف اصلی انتقال از نمونه‌های اولیه به دستگاه‌های تجاری کاملاً عملیاتی و در دسترس بازار است. در این مرحله، فناوری‌های کوانتومی حسگری باید به بلوغ کافی رسیده باشند تا بتوانند به‌صورت گسترده در صنایع مختلف از پزشکی و زیست‌شناسی گرفته تا ناوبری و اکتشافات معدنی به کار گرفته شوند.

مقاله علمی مروری

DevOps

فرهنگ کاری، اهمیت سازمانی و چشم‌انداز شغلی و درآمدی

نویسنده: محمد مهدی عباس زاده

چکیده

خلاصه مقاله

ترکیبی از فرهنگ سازمانی، فرایندهای کاری و ابزارهای فنی است که (DevOps) دواپس با هدف کوتاه‌کردن چرخه توسعه و افزایش کیفیت و پایداری محصولات نرم‌افزاری شکل گرفته است. این مقاله مفهوم دواپس، مبانی فرهنگی آن، اهمیتش در صنعت فناوری اطلاعات، مهارت‌ها و ابزارهای کلیدی، و وضعیت بازار کار و درآمد متخصصان این حوزه را بررسی می‌کند. یافته‌ها نشان می‌دهد موفقیت در دواپس بیش از آنکه به ابزار وابسته باشد، به تغییر نگرش و همکاری میان تیم‌ها وابسته است.

مقدمه

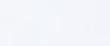
در دهه‌های اخیر، سرعت تغییر در نیازهای کسب‌وکار و انتظارات مشتریان، سازمان‌های فناوری اطلاعات را به بازنگری در شیوه تولید و ارائه نرم‌افزار واداشته است. به‌طور سنتی، تیم مسئول اجرا و (Operations) مسئول نوشتن کد و تیم عملیات (Development) توسعه، نگهداری آن در محیط تولید بود؛ جدایی این دو حوزه اغلب به تأخیر در انتشار نسخه‌ها افزایش خطاهای انسانی و کاهش کیفیت خدمات منجر می‌شد.

است که برای نخستین‌بار Operations و Development دواپس واژه‌ای برساخته از ترکیب در اواخر دهه ۲۰۰۰ میلادی مطرح شد و امروزه به یکی از ارکان اصلی صنعت نرم‌افزار تبدیل شده است. هدف این مقاله، ارائه تصویری جامع از مفهوم، فرهنگ، اهمیت و جنبه‌های شغلی دواپس است.

DevOps تعریف و مبانی

دواپس را می‌توان مجموعه‌ای از اصول، شیوه‌ها و ابزارهایی تعریف کرد که همکاری میان تیم‌های توسعه نرم‌افزار و عملیات فناوری اطلاعات را افزایش می‌دهد تا فرایند ساخت آزمایش و انتشار نرم‌افزار سریع‌تر، قابل‌اعتمادتر و خودکارتر شود. برخلاف تصور رایج، دواپس، صرفاً یک فناوری یا ابزار خاص نیست، بلکه فلسفه‌ای است که در تقاطع سه عنصر افراد، فرایند و فناوری قرار می‌گیرد.





► اصول بنیادین

- یکپارچه‌سازی و تحویل مستمر (CI/CD): خودکارسازی فرایند ساخت، آزمایش و استقرار نرم‌افزار.
- زیرساخت به‌مثابه کد (Infrastructure as Code): مدیریت زیرساخت از طریق کد نسخه‌بندی‌شده.
- پایش و بازخورد مستمر (Monitoring & Feedback): جمع‌آوری داده از تولید برای بهبود محصول.
- خودکارسازی (Automation): کاهش خطای انسانی و افزایش سرعت با حذف کارهای تکراری.
- امنیت یکپارچه (DevSecOps): در نظر گرفتن امنیت از همان مراحل ابتدایی توسعه.

فرهنگ کاری در DevOps

شاید مهم‌ترین و در عین حال دشوارترین بخش پیاده‌سازی دواپس، تغییر فرهنگ سازمانی باشد. بسیاری از پروژه‌های دواپس با وجود بهره‌گیری از پیشرفته‌ترین ابزارها، به دلیل نبود فرهنگ همکاری با شکست مواجه می‌شوند.

► مسئولیت مشترک

در فرهنگ دواپس، مرز میان «نوشتن کد» و «اجرای آن در تولید» کمرنگ می‌شود. این رویکرد اغلب با عبارت «شما آن را ساختید، شما آن را اجرا می‌کنید» توصیف می‌شود.

► فرهنگ یادگیری از شکست

در سازمان‌های دواپسی، خطا فرصتی برای یادگیری تلقی می‌شود. جلسات «بازبینی بدون سرزنش» پس از هر حادثه برگزار می‌شود تا علت ریشه‌ای مشکل شناسایی و از تکرار آن جلوگیری شود.

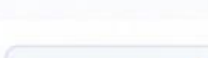
► همکاری و ارتباط باز

ارتباط شفاف میان تیم‌های مختلف—از توسعه و آزمون گرفته تا امنیت و پشتیبانی—یکی از پیش‌نیازهای اصلی دواپس است.

► بهبود مستمر

چرخه بازخورد

برنامه‌ریزی ← ساخت ← آزمایش ← انتشار ← بهره‌برداری ← پایش، و بازگشت به برنامه‌ریزی؛ چرخه‌ای بی‌وقفه که محصول را به‌طور تدریجی بهبود می‌بخشد.





اهمیت DevOps در سازمان‌ها

اهمیت دواپس را می‌توان از چند زاویه بررسی کرد: کسب‌وکاری، فنی و انسانی. سازمان‌هایی که فرهنگ دواپس را پذیرفته‌اند، معمولاً قادرند نسخه‌های جدید محصول را با فاصله زمانی کوتاه‌تر و نرخ خطای پایین‌تر منتشر کنند.

- کاهش زمان عرضه به بازار (Time to Market) و افزایش چابکی سازمانی.
- افزایش پایداری و در دسترس بودن خدمات از طریق پایش و بازیابی خودکار.
- کاهش هزینه‌های عملیاتی با خودکارسازی فرایندهای تکراری.
- افزایش رضایت مشتری از طریق ارائه سریع‌تر ویژگی‌ها و رفع سریع‌تر خطاها.
- ایجاد هم‌راستایی میان اهداف فنی و اهداف کسب‌وکار.

از منظر فنی نیز دواپس بستری فراهم می‌کند تا سازمان‌ها از معماری‌های نوینی مانند میکروسرویس‌ها، کانتینرها و زیرساخت ابری بهره ببرند.

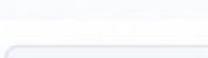
مهارت‌ها و ابزارهای کلیدی

متخصص دواپس معمولاً ترکیبی از دانش برنامه‌نویسی، سیستم‌عامل، شبکه و ابزارهای خودکارسازی را در اختیار دارد:

- کنترل نسخه: Git و پلتفرم‌هایی مانند GitHub و GitLab.
 - یکپارچه‌سازی و تحویل مستمر: Jenkins، GitLab CI/CD، GitHub Actions.
 - کانتینرسازی و ارکستراسیون: Docker و Kubernetes.
 - زیرساخت به‌مثابه کد: Terraform، Ansible و CloudFormation.
 - پایش و لاگ‌برداری: Prometheus، Grafana و پشته ELK.
 - سرویس‌های ابری: AWS، Google Cloud و Microsoft Azure.
- علاوه بر دانش فنی، مهارت‌های نرم مانند حل مسئله، کار تیمی و ارتباط مؤثر نیز در این حوزه اهمیت بالایی دارند.

بازار کار و درآمد متخصصان DevOps

با گسترش سریع خدمات دیجیتال و مهاجرت سازمان‌ها به زیرساخت ابری، تقاضا برای متخصصان دواپس به‌طور پیوسته افزایش یافته است. این حوزه معمولاً در فهرست مشاغل پرتقاضا و پردرآمد صنعت فناوری اطلاعات قرار می‌گیرد، چرا که در تقاطع چند تخصص فنی قرار دارد.





► سطوح رشد شغلی و درآمدی

- سطح مبتدی (Junior): آشنایی با مبانی لینوکس، شبکه، اسکریپت‌نویسی و ابزارهای پایه CI/CD.
- سطح میانی (Mid-level): تسلط بر کانتینرسازی، ارکستراسیون و مدیریت زیرساخت ابری.
- سطح ارشد (Senior/Lead): طراحی معماری زیرساخت، مدیریت امنیت و رهبری فنی تیم.

نکته کاربردی

بخش قابل‌توجهی از متخصصان دواپس امکان همکاری دورکار با کارفرمایان خارجی را دارند که می‌تواند درآمد را نسبت به بازار محلی به‌طور محسوسی افزایش دهد؛ اما این مسیر معمولاً به تسلط بر انگلیسی فنی، نمونه‌کار عملی و گواهینامه‌های تخصصی نیاز دارد (AWS، Kubernetes، Terraform).

چالش‌ها و چشم‌انداز آینده

با وجود مزایای فراوان، پیاده‌سازی دواپس با چالش‌هایی همراه است؛ از جمله مقاومت فرهنگی در برابر تغییر، پیچیدگی فنی ابزارها، هزینه اولیه آموزش و نیاز به هماهنگی مستمر میان تیم‌ها. با رشد هوش مصنوعی، مفاهیمی مانند AIOps و پلتفرم‌مهندسی (Platform Engineering) در حال شکل‌گیری‌اند که می‌توانند آینده این حوزه را متحول کنند.

نتیجه‌گیری

دواپس بیش از آنکه صرفاً مجموعه‌ای از ابزارها باشد، تغییری بنیادین در فرهنگ سازمانی و شیوه همکاری میان تیم‌های توسعه و عملیات است. از منظر شغلی نیز، دواپس با توجه به ماهیت میان‌رشته‌ای و تقاضای رو به رشد بازار، همچنان یکی از مسیرهای شغلی جذاب و پردرآمد در صنعت فناوری اطلاعات به شمار می‌رود.

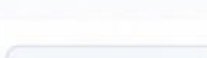
منابع

Kim, G., Humble, J., Debois, P., & Willis, J. (2016). The DevOps Handbook. IT Revolution Press.

Forsgren, N., Humble, J., & Kim, G. (2018). Accelerate: The Science of Lean Software and DevOps. IT Revolution Press.

Amazon Web Services. (n.d.). What is DevOps? AWS.

Microsoft Azure. (n.d.). DevOps solutions documentation.



انقلاب در دنیای بازی سازی؛ از رونمایی از Unreal Engine 6 تا تحولات هوش مصنوعی در نسخه 5.8

در جریان جشنواره شیکاگو، شرکت Epic Games از چشم انداز آینده خود پرده برداشت؛ از معرفی نسل جدید موتور

Unreal Engine 6 تا انتشار به روزرسانی های حیاتی نسخه 5.8 که مرز میان خلاقیت انسانی و هوش مصنوعی را جابه جا می کند.

Unreal Engine 6: فراتر از یک موتور توسعه، یک اکوسیستم زنده

شرکت Epic اعلام کرد که نسل بعدی موتور توسعه بازی خود، یعنی Unreal Engine 6، فراتر از ارتقای گرافیک خواهد بود. این موتور قصد دارد قابلیت های سطح AAA نسخه ۵ را با فرآیندهای تولید زنده (Live Production) که در Fortnite آزمایش شده اند، ترکیب کند.

هدف اصلی Epic، ایجاد بستری است که توسعه دهندگان بتوانند تجربه هایی با «هر مقیاس و دامنه» را تنها یک بار تولید کنند و سپس آن ها را به راحتی میان پلتفرم های سنتی، اکوسیستم Fortnite و دنیای بازی های آنلاین و چندمحصولی اختصاصی خود، منتشر کنند.

Unreal Engine 6 بر سه ستون اصلی استوار خواهد بود:

1. انتقال به زبان Verse: با جایگزینی یا تکمیل مدل های فعلی، زبان برنامه نویسی Verse با ارائه رویکردی «تراکنشی» (Transactional) در کنار ++C، دسترسی به توسعه بازی را آسان تر

این امر ساخت تجربه های زنده، پایدار و مقیاس پذیر را ممکن می سازد که هزاران کاربر می توانند به طور هم زمان در آن ها مشارکت کنند.

2. اقتصاد و محتوای متقابل: از طریق استانداردهای باز، امکان جابه جایی محتوا، کدها و اقتصادهای درون بازی



(In-game Economies) میان بازی‌ها و اکوسیستم‌های مختلف فراهم خواهد شد؛ امری که همکاری میان توسعه‌دهندگان را در ابعادی بی‌سابقه قرار می‌دهد.

1. زنجیره تولید هوشمند (AI-Driven Pipeline): با بهره‌گیری از پروتکل متن‌باز مدل (MCP)، این موتور با هوش مصنوعی‌هایی نظیر Gemini و Claude یکپارچه می‌شود. هدف این است که کارهای تکراری و زمان‌بر به هوش مصنوعی سپرده شود تا تیم‌های توسعه بتوانند تمام تمرکز خود را بر جنبه‌های خلاقانه و فنی قرار دهند.

زمان دسترسی زودهنگام (Early Access) به Unreal Engine 6 برای پایان سال ۲۰۲۷ برنامه‌ریزی شده است.

Unreal Engine 5.8: بلوغ ابزارها و آماده‌سازی برای تولید انبوه
در نسخه ۵.۸، بسیاری از قابلیت‌های کلیدی به مرحله «آماده برای تولید» (Production Ready) رسیده‌اند. از جمله این ابزارهای قدرتمند می‌توان به موارد زیر اشاره کرد:

• MegaLights و Chaos Cloth

• Hub Audio برای Live Link

• Insights و Iris ،Dataflow

• Movie Render Graph

نوآوری‌های فنی در این نسخه:

- سیستم Terrain Mesh: این سیستم جدید اجازه می‌دهد بدون محدودیت‌های سنتی «میدان ارتفاع» (Heightfield)، مناظر سه‌بعدی پیچیده در هر نوع محیطی ساخته شود.
- بهینه‌سازی خیره‌کننده: بهینه‌سازی فرآیند کامپایل Shader و سیستم حذف داده‌های تکراری (Deduplication)، باعث کاهش ۶۸ درصدی تعداد Shaderهای مورد نیاز در Fortnite شده است.



• نورپردازی پویا برای کنسول‌های نسل جدید: سیستم Lumen اکنون با رویکردی سبک‌تر و پویا، امکان اجرای بازی با نرخ ۶۰ فریم بر ثانیه را روی Nintendo Switch 2 و رایانه‌های شخصی فراهم می‌کند.

ادغام عمیق هوش مصنوعی؛ هوش مصنوعی دیگر فقط یک دستیار نیست!

یکی از هیجان‌انگیزترین بخش‌های نسخه ۵.۸، معرفی پلاگین آزمایشی برای پروتکل MCP است. این قابلیت به توسعه‌دهندگان اجازه می‌دهد مدل‌های هوش مصنوعی (مانند Claude) را مستقیماً به پروژه‌های خود متصل کنند.

در این رویکرد، هوش مصنوعی دیگر صرفاً ابزاری برای نوشتن کد یا توضیح مفاهیم نیست؛ بلکه به یک «همکار فعال» تبدیل می‌شود که گردش کارهای پیچیده درون Unreal Engine را درک کرده و در اجرای آن‌ها مشارکت می‌کند. توسعه‌دهندگان در انتخاب مدل (Claude، Gemini یا غیره) کاملاً آزاد هستند.

انقلاب در تولید تصویر و ویدئو:

با ورود مدل‌های Diffusion به محیط Unreal، هنرمندان کنترل بی‌نظیری بر خروجی‌های خود خواهند داشت. برخلاف ابزارهای مبتنی بر متن ساده، در این نسخه می‌توان از داده‌های سه‌بعدی مانند نقشه عمق (Depth Map)، نقشه نرمال (Normal Map) و اطلاعات دوربین به‌عنوان ورودی استفاده کرد.

این یعنی:

- تولید فریم‌هایی که دقیقاً با کادربندی و چیدمان صحنه سه‌بعدی شما مطابقت دارند.
- استخراج اشیاء سگمنت‌شده و تبدیل آن‌ها به دارایی‌های (Assets) سه‌بعدی قابل استفاده.
- تولید مستقیم ویدئوهای باکیفیت در داخل خود موتور بازی‌سازی.

نویسنده : کیمیا رهنا آذر



پادکست چرا کامپیوتر؟

اولین پادکست نفوتک با یک سؤال ساده آغاز می‌شود:

چرا کامپیوتر؟

در این گفت‌وگو تلاش می‌کنیم فراتر از پاسخ‌های معمولی مثل بازار کار و آینده فناوری، به این سؤال نگاه کنیم که کامپیوتر چه چیزی به ما می‌دهد و چرا یادگیری آن اهمیت دارد.

کامپیوتر چیست؟

در پادکست درباره گستردگی دنیای کامپیوتر صحبت می‌کنیم؛ از سخت‌افزار و نرم‌افزار گرفته تا شبکه، امنیت، پایگاه داده و هوش مصنوعی. کامپیوتر دیگر فقط یک دستگاه نیست و تقریباً در تمام صنایع و بخش‌های زندگی حضور دارد.

کامپیوتر؛ فقط برنامه‌نویسی نیست

یکی از موضوعات اصلی، **حل مسئله و تفکر محاسباتی (Computational Thinking)** است. درباره تحلیل و تجزیه مسائل، طراحی الگوریتم و اهمیت پیچیدگی زمانی صحبت می‌کنیم و نشان می‌دهیم که علوم کامپیوتر فراتر از صرفاً کدنویسی است.

مسیرهای مختلف کامپیوتر

در ادامه به حوزه‌هایی مانند شبکه، امنیت سایبری، داده، یادگیری ماشین، معماری کامپیوتر و هوش مصنوعی می‌پردازیم و این سؤال را بررسی می‌کنیم که آیا نداشتن علاقه به برنامه‌نویسی به معنی مناسب نبودن رشته کامپیوتر است یا خیر.

کامپیوتر در دنیای واقعی

از نقش کامپیوتر در **بانکداری، پزشکی، صنعت، حمل‌ونقل و سرگرمی** صحبت می‌کنیم و در کنار قدرت فناوری، موضوعاتی مانند حریم خصوصی، امنیت اطلاعات و مسئولیت استفاده از داده‌ها را نیز بررسی می‌کنیم.

دانشگاه و آینده دانشجویان

بخش دیگری از پادکست به نقش دانشگاه و اهمیت یادگیری مفاهیم پایه مانند الگوریتم، ساختمان داده، شبکه، سیستم عامل و پایگاه داده اختصاص دارد. همچنین بر **پروژه، تمرین، تجربه واقعی و یادگیری مداوم** تأکید می‌شود.

پس چرا کامپیوتر؟

در پایان به این نتیجه می‌رسیم که کامپیوتر فقط یک ابزار فنی نیست؛ بلکه راهی برای **فکر کردن، حل مسئله و تبدیل ایده به واقعیت** است. شاید مهم‌ترین سؤال بعد از شنیدن این پادکست دیگر «چرا کامپیوتر؟» نباشد، بلکه این باشد:

من با کامپیوتر، چه چیزی می‌توانم بسازم؟



برای مشاهده نسخه کامل
پادکست اسکن کنید یا به
سایت نفوتک مراجعه کنید

نگاهی کوتاه به مهم ترین های کامیک کان سن دیگو ۲۰۲۶

نویسنده: کیمیا رهنما آذر

یکی از بزرگترین آخر هفته های دنیای سرگرمی که برگزار شد

کامیک کان سن دیگو ۲۰۲۶، یکی از بزرگترین رویدادهای دنیای سرگرمی، امسال هم پر از غافلگیری های بزرگ بود. از رونمایی پوستر و کلیپ جدید Avengers: Doomsday با حضور ترسناک ویکتور فون دوم و ارتش سنتینل ها، تا تأیید رسمی تاریخ انتشار Black Panther 3 و معرفی دیوید جانسون به عنوان پلنگ سیاه جدید. در دنیای سریال ها نیز خبرهای داغی داشتیم؛ از اولین تریلر Blade Runner 2099 با فضایی روشن تر و متفاوت، تا فصل سوم The Lord of the Rings: The Rings of Power که برای اولین بار بالروگ را به سخن می آورد. در بخش بازی های ویدیویی نیز Marvel's Wolverine با تریلری جدید و تاریخ انتشار مشخص، یکی از مهم ترین عناوین سال ۲۰۲۶ معرفی شد، در ادامه با ما همراه باشید.

نمایش و پوستر جدید فیلم Avengers: Doomsday در کامیک کان سن دیگو رونمایی شد

در جدیدترین رویداد کامیک کان سن دیگو، مارول استودیوز از پوستر جدید فیلم انتقام جویان: روز رستاخیز (Avengers: Doomsday) رونمایی کرد. در این پوستر، «ویکتور فون دوم» با چهره ای اسرارآمیز و سایه زده، جلوتر از همه در حال حرکت دیده می شود که حالتی ترسناک تر از همیشه دارد.

اما مهم ترین بخش رویداد، نمایش کلیپی بود که در آن دکتر دوم ارتشی از ربات های خفته «سنتینل» را به حرکت درمی آورد؛ صحنه ای که به گفته حاضران، یکی از وفادارترین لحظات به کمیک های اصلی در کل تاریخ سینمایی مارول محسوب می شود.



همچنین نماهایی از شخصیت‌هایی همچون ثور، شانگ چی، شوری و گامبیت به نمایش درآمد. در بخش دیگری از این رویداد، کلیپ طنزی از مبارزه پانیشر و اسپایدرمن با هالک پخش شد که این نسخه از هالک، ترکیبی از خشم وحشی و توانایی گفت‌وگوی منطقی توصیف شده است.

«Avengers: Doomsday» در تاریخ ۱۸ دسامبر ۲۰۲۶ (۲۷ آذر ۱۴۰۵) اکران می‌شود.

تاریخ انتشار Black Panther 3 توسط مارول تأیید شد، دیوید جانسون در نقش پلنگ سیاه جدید انتخاب شد.

تاریخ رسمی اکران: فیلم بلک پنتر
۳ رسماً توسط مارول تأیید شده و تاریخ
اکران آن ۱۵ دسامبر ۲۰۲۸ (۲۵ آذر
۱۴۰۷) اعلام شده است.

بازیگر نقش جدید بلک پنتر: دیوید
جانسون بازیگر فیلمهای بیگانه:
رومولوس و قدم طولانی (در نقش پسر
تی‌چالا که در فیلم دوم به صورت کودکی
دیده شد) انتخاب شده است و قرار است
عنوان بلک پنتر را به ارث ببرد. بازگشت
بازیگران اصلی: لتیشیا رایت در
نقش شوری و وینستون دوک در
نقش ام‌باکو به این فیلم بازمی‌گردند.

حضور دنزل واشنگتن: رایان
کوگلر (کارگردان) تأیید کرده که نقش
خاصی برای دنزل واشنگتن در فیلم
نوشته است و او در این پروژه حضور
خواهد داشت



آینده‌ای روشن در سریال Blade Runner 2099

سریال بلید رانر ۲۰۹۹ در ۲۵ نوامبر ۲۰۲۶ (۴ آذر ۱۴۰۵) با تمام ۸ قسمت به طور همزمان پخش می‌شود. داستان ۵۰ سال پس از فیلم بلید رانر ۲۰۴۹ رخ می‌دهد. در این زمان، ربات‌ها (ریپلیکانت‌ها) بر انسان‌ها پیروز شده‌اند و اکنون حاکم شهر لس‌آنجلس هستند و انسان‌ها شهروندان درجه دو محسوب می‌شوند. داستان دربارهٔ کورا (با بازی هانتز شیفر)، یک انسان فراری است که خود را به جای یک «بلید رانر» (شکارچی ربات‌ها) جا می‌زند و مجبور می‌شود با اولون (با بازی میشل یو)، یک بلید رانر ربات که روزهای پایانی عمرش را می‌گذراند، همکاری کند تا ربات فراری را پیدا کنند. ریدلی اسکات (کارگردان فیلم اصلی) به عنوان تهیه‌کننده اجرایی در این پروژه حضور دارد.



اولین نگاه به فصل سوم The Lord of the Rings: The Rings of Power

تغییر بزرگ در روایت: بالروگ که در آثار تالکین هرگز حرف نمی‌زند، در فصل سوم سریال حرف خواهد زد.

سازندگان سریال می‌گویند می‌خواهند «جنبه متفاوتی از او» را نشان دهند و او را یک «هوشمند و خطرناک» به تصویر بکشند. **اتحاد الف‌ها و دورف‌ها در این فصل سوم** بالروگ در این فصل با «سائورون» صحبت خواهد کرد و سائورون هنر ساخت

«حلقه یگانه» را از او یاد می‌گیرد. **تاریخ پخش:** فصل سوم از ۱۱ نوامبر ۲۰۲۶ (۲۰ آبان ۱۴۰۵) پخش می‌شود.



Marvel's Wolverine

بازی در ۱۵ سپتامبر ۲۰۲۶ (۲۴ شهریور ۱۴۰۵) به صورت انحصاری برای پلی‌استیشن ۵ عرضه می‌شود و یک تریلر داستانی جدید نیز رونمایی شد.

داستان در دنیایی می‌گذرد که (Team X) جایگزین (X-Men) شده است این گروه در دهه ۱۹۶۰ توسط «ناتانیل اسکس» برای مقابله با تهدید «بولیوار ترسک» تشکیل شده هدف ترسک، نابودی همهٔ جهش‌یافتگان استسه سال قبل از داستان بازی، تیم ایکس را ترک کرده و حالا بازگشته است **جین گری** به تنهایی به دنبال جهش‌یافتگان دیگر می‌گردد تا پناهگاهی امن برایشان فراهم کند و در این مسیر با لوگان آشنا می‌شود.



سخن بزرگان

«ما فقط می‌توانیم فاصله کوتاهی از آینده را ببینیم، اما در همین فاصله کارهای بسیاری وجود دارد که باید انجام شوند.»

آلن تورینگ؛ پیشگام علوم کامپیوتر

آلن تورینگ ریاضی‌دان و دانشمند بریتانیایی و از پایه‌گذاران علوم کامپیوتر مدرن بود. او با ارائه مفهوم ماشین تورینگ، نقش مهمی در شکل‌گیری نظریه محاسبات داشت.

تورینگ در جنگ جهانی دوم در رمزگشایی Enigma نقش مهمی ایفا کرد و بعدها با مطرح کردن آزمون تورینگ، از نخستین افرادی بود که به‌طور جدی درباره امکان تفکر ماشین‌ها و هوش مصنوعی اندیشید.



نئوتک

رسانه فناوری های نوین

نئوتک، فصلنامه ای از دل دانشگاه برای آینده ای هوشمندتر
روایتی از علم، خلاقیت، دانش و نوآوری در دنیای بی پایان فناوری



بازی



اخبار



طنز



آموزشی



هوش
مصنوعی



پادکست



مقاله علمی



مصاحبه



دورنگاه

“ فناوری، فقط ابزار نیست؛
سبک زندگی، آینده و انتخاب ماست. ”



نسخه دیجیتال نشریه
اسکن کنید و بخوانید

با نئوتک، آینده را بسازیم...