



# OPEN SOURCE Technologies<sup>1st</sup> Edition

## معرفی کوتاه کتاب

این کتاب تلاشی جامع و تحلیلی است برای معرفی، بررسی ابعاد مختلف فناوری‌های متن باز، از ریشه‌های تاریخی و فلسفی آن گرفته تا نقش کلیدی‌اش در هوش مصنوعی، اینترنت اشیا، امنیت سایبری، و صنایع گوناگون. با نگاهی دقیق به مفاهیمی مانند تفاوت نرم‌افزار آزاد و است. این کتاب، OSI و FSF متن باز، مجوزهای نرم‌افزاری، نقش بنیادهای تأثیرگذار همچون برای توسعه‌دهندگان، مدیران فناوری، دانشگاهیان و تمامی علاقه‌مندان به آینده‌ی باز فناوری نوشته شده است. اگر به دنبال درکی عمیق، ساختاریافته و به‌روز از اکوسیستم متن باز هستید، این کتاب نقطه‌ی شروعی قدرتمند است

کتاب  
مرجع  
تکنولوژی  
متن باز

تکنولوژی های متن باز

# Open Source Technologies

حسین سیلانی

ویرایش اول



|                         |                                               |
|-------------------------|-----------------------------------------------|
| سرشناسه                 | سیلانی، حسین، ۱۳۶۰،                           |
| عنوان و نام پدیدآور     | تکنولوژی های متن باز Open Source Technologies |
| مشخصات نشر              | حسین سیلانی؛ ویراستاری حسین سیلانی.           |
| مشخصات ظاهری            | تهران: کیان دانش، ۱۴۰۴.                       |
| شابک                    | ۲۷۰ ص.                                        |
| وضعیت فهرست نویسی       | ۹۷۸، ۶۲۲، ۴۰۰، ۱۵، ۶                          |
| موضوع                   | فیپا                                          |
|                         | سیستم عامل                                    |
|                         | Operating System                              |
| موضوع                   | سیستم های عامل (کامپیوتر)                     |
|                         | Operating systems (Computers)                 |
| شناسه افزوده            | سیلانی، حسین، ۱۳۶۰،                           |
| رده بندی کنگره          | ۷۶/۷۶QA                                       |
| رده بندی دیویی          | ۴۳۲/۰۰۵                                       |
| شماره کتابشناسی ملی     | ۹۶۷۳۵۹۳                                       |
| اطلاعات رکورد کتابشناسی | فیپا                                          |

نشر کیان دانش: ناشر کتاب های دانشگاهی



عنوان: فلسفه یونیکس  
تالیف: حسین سیلانی  
صفحه آرای و ویراستاری: حسین سیلانی  
چاپ و صحافی: موسسه مهراد  
طراحی جلد: حسین سیلانی  
چاپ اول: ۱۴۰۴  
شمارگان: ۱۰۰ جلد  
شابک: ۶، ۰۱۵، ۴۰۰، ۶۲۲، ۹۷۸  
حق چاپ محفوظ است.

خیابان ۱۲ فروردین، خیابان شهدای ژاندارمری، شماره ۱۲۶، طبقه چهارم

سخنی از نویسنده  
با سلام و احترام به دوست گرامی

از اینکه این کتاب را انتخاب کرده‌اید و در این مسیر همراه من هستید صمیمانه سپاسگزارم. امیدوارم این اثر برای علاقه‌مندان به حوزه لینوکس مفید و اثربخش باشد. همچنین از شما درخواست دارم که اگر نواقص یا کاستی‌هایی در کتاب مشاهده کردید آن‌ها را نادیده نگیرید و با ارائه نظرات و پیشنهادات سازنده‌ی خود، به بهبود کیفیت این اثر کمک کنید.

شایان ذکر است که این کتاب یکی از صدها کتاب لینوکسی من است که بسیاری از آن‌ها در حال ترجمه یا نگارش هستند. این مجموعه در راستای پروژه‌های متن‌باز شخصی‌ام و با هدف طراحی و توسعه توزیع‌های لینوکس نوشته شده است. امیدوارم این تلاش‌ها گامی هرچند کوچک در جهت گسترش دانش و استفاده از سیستم‌عامل‌های متن‌باز باشد.

🔗 <https://predator.os.ir/>

🔗 <https://little.Pscho.ir>

🔗 <https://emperor.os.ir/>

🔗 <https://ubuntu.ir>

تصمیم دارم محتوای آموزشی فارسی در حوزه‌های لینوکس و نرم‌افزارهای متن‌باز را در قالب‌های متنوعی مانند کتابچه، کتاب، فلش‌کارت، فایل‌های صوتی و ویدئوهای آموزشی تولید کرده و از طریق یک سایت در دسترس عموم قرار دهم. هدف من از این کار، ایجاد یکی از بزرگترین و جامع‌ترین مراجع آموزشی فارسی در این زمینه است. برای دسترسی به سایت و اطلاع از آخرین اخبار و محتواهای آموزشی، می‌توانید از طریق راه‌های ارتباطی زیر با من در تماس باشید:

🔗 آدرس سایت: [learninghive.ir](http://learninghive.ir)

🔗 کانال تلگرام: @Linuxtnt

🔗 آیدی تلگرام: @seilany

🔗 ایمیل: [h.seilany@gmail.com](mailto:h.seilany@gmail.com)

🔗 گیت‌هاب: [hosseinseilani](https://github.com/hosseinseilani)

منتظر حضور و همراهی شما هستم!

تقديم به تو

يگانه فرزانه دلم

## فهرست مطالب

### مقدمه - ۱۹ -

- ۱۹ - چه کسانی باید این کتاب را بخوانند؟
- ۲۰ - تکنولوژی متن باز چیست؟
- ۲۲ - تاریخچه تکنولوژی‌های متن باز
- ۲۳ - ویژگی‌های کلیدی تکنولوژی‌های متن باز
- ۲۴ - مزایای تکنولوژی‌های متن باز
- ۲۶ - چرا تکنولوژی‌های متن باز مهم هستند؟
- ۲۷ - تفاوت با نرم‌افزار آزاد (Free Software)

### فصل ۱ - ۲۹ -

#### منبع باز - ۲۹ -

- ۲۹ - چگونه به اینجا رسیدیم
- ۲۹ - مزایای فناوری‌های منبع باز
- ۳۰ - چالش‌های فناوری‌های منبع باز
- ۳۰ - وضعیت فعلی فناوری‌های منبع باز
- ۳۰ - بنیاد نرم‌افزار آزاد (Free Software Foundation)
- ۳۱ - دستاوردهای مهم آزمایشگاه بل
- ۳۱ - بنیاد نرم‌افزار آزاد (Free Software Foundation) یا
- ۳۲ - اهداف و مأموریت‌های بنیاد نرم‌افزار آزاد
- ۳۲ - چهار آزادی نرم‌افزار آزاد
- ۳۲ - پروژه‌های تحت حمایت FSF
- ۳۲ - فعالیت‌های FSF
- ۳۲ - تأثیر FSF بر دنیای فناوری
- ۳۳ - اهداف و فعالیت‌های بنیاد نرم‌افزار آزاد
- ۳۳ - پروژه گنو و نقش آن در جنبش نرم‌افزار آزاد
- ۳۳ - تفاوت بین نرم‌افزار آزاد و متن باز

- ۳۳- \_\_\_\_\_ تعریف نرم افزار آزاد (Free Software)
- ۳۴- \_\_\_\_\_ تعریف متن باز (Open Source)
- ۳۴- \_\_\_\_\_ تفاوت های فلسفی
- ۳۴- \_\_\_\_\_ تفاوت های عملی
- ۳۴- \_\_\_\_\_ مثال هایی از نرم افزار آزاد و متن باز
- ۳۵- \_\_\_\_\_ نقش ریچارد استالمن در جنبش نرم افزار آزاد
- ۳۵- \_\_\_\_\_ فلسفه نرم افزار آزاد از دیدگاه ریچارد استالمن
- ۳۵- \_\_\_\_\_ تأثیر ریچارد استالمن بر جامعه فناوری
- ۳۶- \_\_\_\_\_ چالش ها و انتقادات
- ۳۶- \_\_\_\_\_ عصر لینوکس (Linux)
- ۳۶- \_\_\_\_\_ تأثیر ۸۰۳۸۶ بر صنعت سخت افزار و نرم افزار
- ۳۷- \_\_\_\_\_ ظهور لینوکس و نقش آن در عصر جدید
- ۳۷- \_\_\_\_\_ تأثیر لینوکس بر صنعت فناوری
- ۳۷- \_\_\_\_\_ لینوکس و جنبش نرم افزار آزاد
- ۳۸- \_\_\_\_\_ چالش ها و آینده لینوکس
- ۳۸- \_\_\_\_\_ تولد لینوکس (Linux) در سال ۱۹۹۱
- ۳۹- \_\_\_\_\_ نقش مینیکس (Minix) در توسعه لینوکس
- ۳۹- \_\_\_\_\_ یوزنت و نقش آن در گسترش لینوکس
- ۳۹- \_\_\_\_\_ تأثیر لینوکس بر دنیای فناوری
- ۳۹- \_\_\_\_\_ چالش ها و آینده لینوکس
- ۴۱- \_\_\_\_\_ رویکرد کلیسای جامع
- ۴۱- \_\_\_\_\_ رویکرد بازار
- ۴۲- \_\_\_\_\_ مقایسه لینوکس و گنو
- ۴۲- \_\_\_\_\_ تأثیر رویکرد بازار
- ۴۲- \_\_\_\_\_ ابتکار منبع باز (Open Source)

- ۴۳ - انتخاب اصطلاح متن باز (Open Source)
- ۴۳ - تأسیس ابتکار منبع باز (OSI – Open Source Initiative)
- ۴۳ - اهداف OSI
- ۴۴ - تأثیر OSI بر جامعه نرم افزار
- ۴۴ - نقش OSI در توسعه نرم افزارهای منبع باز
- ۴۴ - پروژه دبیان (Debian) و ارتباط آن با OSI
- ۴۵ - مقایسه منبع باز و OSI
- ۴۵ - نقش OSI در توسعه نرم افزارهای منبع باز
- ۴۵ - تعریف منبع باز (Open Source)
- ۴۷ - توضیحات دقیق معیارها
- ۴۷ - اهمیت نرم افزار منبع باز
- ۴۷ - مقایسه ی OSI (Open Source Initiative) و FSF (Free Software Foundation)
- ۴۸ - منبع باز و منبع بسته
- ۵۰ - انواع دیگر نرم افزار
- ۵۱ - ابهام کلمه Free در زبان انگلیسی
- ۵۱ - اصطلاحات رایج دیگر
- ۵۲ - تفاوت های کلیدی متن باز
- ۵۲ - فرآیند توسعه جامعه نرم افزار منبع باز
- ۵۲ - فورک (Fork)
- ۵۳ - جامعه توسعه دهندگان
- ۵۳ - مشارکت در پروژه های متن باز
- ۵۳ - چالش های توسعه نرم افزار متن باز
- ۵۴ - توسعه منبع باز جامعه محور است
- ۵۴ - مشارکت کنندگان و ایجاد شاخه های کد
- ۵۴ - ابزارهای ارتباطی در جامعه منبع باز



مزیای اصلی پروژه‌های منبع باز - ۵۵ -

تقویت جوامع و اشتراک‌گذاری چالش‌ها - ۵۵ -

کد رفتاری و نقش آن در جامعه - ۵۵ -

## فصل ۲ - ۵۸ -

### منبع باز و استانداردهای باز - ۵۸ -

استانداردها چیستند و چرا به آنها نیاز داریم؟ - ۵۸ -

مفهوم منبع باز و استانداردهای باز - ۵۸ -

نیازهای خاص صنایع و استانداردهای مرتبط - ۵۹ -

نمونه‌های متن باز برای استانداردها - ۵۹ -

استانداردهای باز - ۵۹ -

مفهوم استانداردهای باز و بسته - ۵۹ -

مثال عملی: ارسال سند بین دستگاه‌های مختلف - ۶۰ -

استانداردهای باز در عمل - ۶۰ -

مزیای استانداردهای باز - ۶۰ -

مقایسه فرمت‌های PDF و Adobe Acrobat و Microsoft Word - ۶۰ -

فرمت: PDF استاندارد باز و جهانی - ۶۱ -

فرمت: DOCX استاندارد باز مبتنی بر XML - ۶۱ -

مقایسه PDF و DOCX - ۶۲ -

استانداردهای ساختمان - ۶۲ -

سازمان‌های توسعه استاندارد (SDOs) - ۶۳ -

ساختار سازمان‌های توسعه استاندارد (SDOs) - ۶۳ -

اهمیت همکاری در توسعه استانداردها - ۶۳ -

چالش‌های توسعه استانداردها - ۶۴ -

فرآیند توسعه استانداردها - ۶۴ -

نقش سازمان‌های تدوین استاندارد (SDO) - ۶۵ -

- ۶۵ - ..... اصول کلیدی یک استاندارد باز
- ۶۵ - ..... چالش‌های توسعه‌ی استانداردهای باز
- ۶۶ - ..... متن باز با استانداردهای باز
- ۶۶ - ..... مثال اول: استانداردهای پخش تلویزیونی
- ۶۶ - ..... مثال دوم: مرورگرهای اینترنتی و سیستم عامل‌ها
- ۶۷ - ..... فایرفاکس
- ۶۷ - ..... استانداردهای باز و نوآوری
- ۶۷ - ..... چالش‌های استانداردهای باز
- ۶۷ - ..... نقش جامعه منبع باز در استانداردسازی
- ۶۹ - ..... SDOهای اصلی از صنایع مختلف
- ۷۲ - ..... اتحادیه بین‌المللی مخابرات
- ۷۳ - ..... بخش ارتباطات رادیویی ITU
- ۷۳ - ..... بخش استانداردسازی مخابرات ITU
- ۷۴ - ..... بخش توسعه ITU
- ۷۴ - ..... ETSI (European Telecommunications Standards Institute)
- ۷۴ - ..... موسسه استاندارد ارتباطات اروپا
- ۷۸ - ..... CableLabs (کیبل لبز)
- ۷۹ - ..... IETF (Internet Engineering Task Force) کارگروه مهندسی اینترنت
- ۸۲ - ..... OpenDaylight (نور روز باز)
- ۸۲ - ..... penStack (پشته باز)
- ۸۴ - ..... ۵GAA
- ۸۵ - ..... فناوری V2X
- ۸۵ - ..... پیاده‌سازی متن باز
- ۸۶ - ..... استانداردها در صنعت مالی
- ۸۸ - ..... بنیاد FINOS

فصل ۳ ----- ۹۲- مجوز نرم افزار ----- ۹۲-

|                                                    |      |
|----------------------------------------------------|------|
| OSI                                                | ۹۲-  |
| مجوزهای کپی‌لفت(Copyleft)                          | ۹۴-  |
| انتخاب مجوز مناسب                                  | ۹۴-  |
| مجوز آپاچی                                         | ۹۶-  |
| پروژه های برجسته ای که از این مجوز استفاده می کنند | ۹۹-  |
| Kubernetes                                         | ۹۹-  |
| Android Open Source Project (AOSP)                 | ۹۹-  |
| مجوز توزیع نرم افزار برکلی (مجوز BSD)              | ۱۰۱- |
| Nginx                                              | ۱۰۲- |
| مجوز عمومی عمومی گنو                               | ۱۰۳- |
| مجوز (GNU General Public License (GPL)             | ۱۰۷- |
| Linux kernel (هسته لینوکس)                         | ۱۰۸- |
| MySQL و MariaDB                                    | ۱۰۸- |
| Drupal                                             | ۱۰۸- |
| نرم افزار Git                                      | ۱۰۹- |
| پلتفرم edX                                         | ۱۰۹- |
| مجوز MIT                                           | ۱۱۰- |
| Godot Engine                                       | ۱۱۱- |
| Keras                                              | ۱۱۱- |
| Lua                                                | ۱۱۲- |
| PuTTY                                              | ۱۱۲- |
| PowerShell                                         | ۱۱۲- |
| مجوز عمومی موزیلا                                  | ۱۱۳- |

|         |                        |
|---------|------------------------|
| - ۱۱۴ - | Bugzilla               |
| - ۱۱۴ - | LibreOffice            |
| - ۱۱۵ - | RabbitMQ               |
| - ۱۱۵ - | مجوز عمومی Eclipse     |
| - ۱۱۶ - | dual licensing         |
| - ۱۱۶ - | strong copyleft        |
| - ۱۱۷ - | مجوز EPL               |
| - ۱۱۷ - | Eclipse Che            |
| - ۱۱۷ - | JUnit                  |
| - ۱۱۷ - | Jetty                  |
| - ۱۱۸ - | تکثیر مجوز             |
| - ۱۱۸ - | license proliferation  |
| - ۱۱۹ - | درک تکثیر مجوزها       |
| - ۱۱۹ - | بررسی انطباق مجوزها    |
| - ۱۲۰ - | دسته‌بندی مجوزها       |
| - ۱۲۱ - | سود بردن از منبع باز   |
| - ۱۲۲ - | اندروید                |
| - ۱۲۲ - | Multi-licensing        |
| - ۱۲۴ - | ساخت یک اکوسیستم       |
| - ۱۲۴ - | پروژه‌های open source  |
| - ۱۲۵ - | PostgreSQL             |
| - ۱۲۶ - | coding                 |
| - ۱۲۶ - | Kubernetes             |
| - ۱۲۷ - | استراتژی‌های کسب و کار |
| - ۱۲۷ - | شهرت برند              |

شرکا ----- ۱۲۸ -

market strategy ----- ۱۲۹ -

خدمات محصول ----- ۱۳۰ -

فصل ۴ ----- ۱۳۴ -

ایمن سازی سیستم های باز ----- ۱۳۴ -

طراحی و معماری ----- ۱۳۶ -

مرحله طراحی و معماری ----- ۱۳۷ -

مرحله پیاده سازی ----- ۱۳۷ -

مرحله تأیید ----- ۱۳۸ -

انتشار ----- ۱۳۹ -

فلسفه توسعه منبع باز ----- ۱۴۰ -

فرآیندها و ابزارها در چرخه عمر توسعه نرم افزار (SDLC – Software Development Life Cycle) ----- ۱۴۰ -

مخازن کد منبع و اسکن امنیتی ----- ۱۴۰ -

دسته بندی ابزارهای DevSecOps ----- ۱۴۱ -

CVSS ----- ۱۴۲ -

سیاست ها و انطباق ----- ۱۴۴ -

Compliance (انطباق) ----- ۱۴۵ -

SOC ۲ (سیستم و کنترل های سازمانی) ----- ۱۴۵ -

نظارت و استانداردهای امنیتی ----- ۱۴۶ -

OWASP ----- ۱۴۷ -

Injection (تزریق) ----- ۱۴۷ -

Insecure Design (طراحی ناامن) ----- ۱۴۸ -

Security Misconfiguration (پیکربندی اشتباه امنیتی) ----- ۱۴۸ -

Vulnerable and Outdated Components (اجزای آسیب پذیر و قدیمی) ----- ۱۴۸ -

Identification and Authentication Failures (خرابی در شناسایی و احراز هویت) ----- ۱۴۸ -

- ۱۴۸ - Software and Data Integrity Failures (نقص نرم افزار و یکپارچگی داده ها)
- ۱۴۹ - Security Logging and Monitoring Failures (خرابی های ثبت و نظارت امنیتی)
- ۱۴۹ - Server-Side Request Forgery یا به اختصار SSRF (جعل درخواست سمت سرور)
- ۱۴۹ - بنیاد امنیت منبع باز (OpenSSF)
- ۱۵۰ - Identifying Security Threats (شناسایی تهدیدات امنیتی)
- ۱۵۰ - Digital Identity Verification (تأیید هویت دیجیتال)
- ۱۵۴ - تغییرات سریع به سمت فناوری های دیجیتال و امنیت منبع باز

## فصل ۵ - اجزای زیرساخت فیزیکی

- ۱۵۹ - اجزای زیرساخت فیزیکی
- ۱۶۵ - سیستم های تأمین
- ۱۶۶ - اجزای زیرساخت مجازی
- ۱۶۸ - پشته مجازی سازی
- ۱۶۹ - مجازی سازی کامل
- ۱۷۰ - پاراویرچوال سازی
- ۱۷۰ - مجازی سازی در سطح سیستم عامل (کانتینر سازی)
- ۱۷۱ - QEMU
- ۱۷۳ - KVM
- ۱۷۳ - virtIO
- ۱۷۴ - libvirt
- ۱۷۴ - کانتینر
- ۱۷۶ - شبکه های نرم افزاری تعریف شده
- ۱۷۶ - شبکه های تعریف شده توسط نرم افزار
- ۱۷۷ - شبکه مجازی باز (OVN) و اجزای آن
- ۱۷۸ - اجزای اصلی OVN

- معماری و کارکرد OVN ----- ۱۸۱ -
- Ceph پلتفرم ذخیره‌سازی منبع باز ----- ۱۸۱ -
- اجزای کلیدی خوشه Ceph ----- ۱۸۲ -
- چرا Ceph اهمیت دارد؟ ----- ۱۸۴ -
- OpenStack (اوپن‌استک) ----- ۱۸۷ -
- مدیریت و ارکستراسیون ----- ۱۹۲ -

## فصل ۶ ----- ۱۹۷ -

### متن باز ----- ۱۹۷ -

- نرم افزار برای فناوری های نو ظهور ----- ۱۹۷ -
- پیاده سازی هوش مصنوعی ----- ۲۰۲ -
- فاز ۱: آماده سازی داده ها ----- ۲۰۲ -
- فاز ۲: دگرگونی داده ها ----- ۲۰۳ -
- فاز ۳: آموزش ----- ۲۰۴ -
- فاز ۴: شبیه سازی، اعتبارسنجی، و اجرا ----- ۲۰۵ -
- فاز ۵: گسترش ----- ۲۰۶ -
- چارچوب و ابزار موتور هوش مصنوعی ----- ۲۰۷ -
- زبان های برنامه نویسی سطح پایین و سطح بالا ----- ۲۰۷ -
- سخت افزار برای بارهای کاری AI/ML/IoT ----- ۲۱۷ -
- کاربرد هوش مصنوعی، یادگیری ماشینی و اینترنت اشیا در کسب و کارها ----- ۲۲۰ -

## فصل ۷ ----- ۲۲۴ -

### فناوری منبع باز در صنعت ----- ۲۲۴ -

- نقش نرم افزار منبع باز در صنایع مختلف ----- ۲۲۴ -
- هوافضا و نرم افزار منبع باز ----- ۲۲۵ -
- تحول کشاورزی با فناوری و نرم افزار منبع باز ----- ۲۲۷ -
- ابتکارات منبع باز در کشاورزی هوشمند ----- ۲۲۹ -
- نقش فناوری منبع باز در صنعت خودروسازی ----- ۲۳۰ -

- ۲۳۱ - ساختار و گروه‌های تخصصی AGI
- ۲۳۳ - تأثیر AGI بر صنعت خودروسازی
- ۲۳۴ - چالش‌ها و آینده AGI
- ۲۳۴ - تحول صنعت بازی با فناوری منبع باز
- ۲۳۶ - گودو: یک موتور بازی منبع باز برجسته
- ۲۳۶ - بنیاد لینوکس و Open ۳D Engine
- ۲۳۷ - مزایا و چالش‌های موتورهای منبع باز در بازی سازی
- ۲۳۸ - فناوری منبع باز در صنعت ساخت و تولید
- ۲۳۹ - گروه‌های کاری OMP و پروژه‌های مرتبط
- ۲۴۰ - پروژه‌های منبع باز در OMP
- ۲۴۱ - زیرپروژه‌های SDS
- ۲۴۲ - توجه ویژه: اعلان قرار گرفتن در معرض Google-Apple
- ۲۴۲ - تأثیر و آینده منبع باز در تولید

## فصل ۸ - رشد و روند منبع باز

- ۲۴۶ - بررسی مخازن برجسته در GitHub و نقش آن‌ها در اکوسیستم منبع باز
- ۲۴۸ - نگاهی به تأثیرگذاران برجسته در GitHub و نقش آن‌ها در دنیای منبع باز
- ۲۵۰ - تحلیل تأثیرگذاری در دنیای منبع باز
- ۲۵۰ - درس‌هایی برای توسعه‌دهندگان و آینده منبع باز
- ۲۵۱ - نگاهی به آمار زبان‌های برنامه‌نویسی محبوب در GitHub
- ۲۵۳ - کاوش در دامنه‌های نوظهور منبع باز: ارزهای دیجیتال، یادگیری ماشینی و بازی
- ۲۵۳ - ارزهای دیجیتال: انقلابی در مبادلات دیجیتال
- ۲۵۴ - یادگیری ماشینی: قدرت تحلیل و پیش‌بینی
- ۲۵۵ - بازی: خلاقیت و سرگرمی در منبع باز
- ۲۵۶ - برنامه تحصیلی و جایگاه منبع باز



- نقش مربیان در آموزش ابتدایی و متوسطه - ۲۵۷ -
- ابزارهای نرم‌افزاری در آموزش منبع باز - ۲۵۷ -
- برنامه‌های صنعت و تشویق به مشارکت در منبع باز - ۲۵۹ -
- راه‌اندازی پروژه‌های متن‌باز - ۲۶۰ -
- درگیر شدن در منبع باز با شرکت‌ها - ۲۶۲ -
- نوآوری‌های آینده و منبع باز - ۲۶۴ -
- وب ۳٫۰ و نقش منبع باز - ۲۶۴ -
- متاورس و تأثیر منبع باز - ۲۶۵ -
- چندتجربه و نقش منبع باز در تحول دیجیتال - ۲۶۶ -
- محاسبات کوانتومی - ۲۶۷ -
- مالی غیرمتمرکز - ۲۶۷ -
- توکن غیرقابل تعویض (NFT) - ۲۶۸ -
- شبکه داده نام‌گذاری شده - ۲۶۹ -



## مقدمه

به کتاب "شروع کار با فناوری‌های متن‌باز: استفاده از فناوری‌های متن‌باز در پروژه‌ها و موارد استفاده واقعی" خوش آمدید. اگر به دنبال این هستید که بدانید متن‌باز چیست، چگونه عمل می‌کند، کدام صنایع به طور فعال درگیر متن‌باز هستند و جنبه‌های امنیتی متن‌باز کدامند، این کتاب برای شما مناسب است.

## چه کسانی باید این کتاب را بخوانند؟

این کتاب بیشتر مناسب شرکت‌ها (توسعه‌دهندگان/اپراتورها/مدیریت) و دانشجویان است که می‌خواهند دیدگاه ۳۶۰ درجه‌ای از متن‌باز به دست آورند، فارغ از اینکه در مراحل اولیه یا پیشرفته‌ای از استفاده از هر فناوری متن‌باز قرار دارند. این کتاب بر اساس پروژه‌های واقعی و موارد استفاده عملی نوشته شده است تا به شما کمک کند اکوسیستم متن‌باز را بهتر درک کنید.

متن‌باز (Open Source) به نرم‌افزارهایی اشاره دارد که کد منبع آنها به صورت عمومی در دسترس است و هر فردی می‌تواند آن را مشاهده، تغییر و توزیع کند. این ویژگی باعث شده است که متن‌باز به یکی از پایه‌های مهم توسعه فناوری در دنیای امروز تبدیل شود. یکی از مزایای اصلی متن‌باز، شفافیت آن است؛ زیرا کدها توسط جامعه گسترده‌ای از توسعه‌دهندگان بررسی می‌شوند، که این امر می‌تواند به بهبود کیفیت و امنیت نرم‌افزار کمک کند.

صنایع مختلفی از جمله فناوری اطلاعات، سلامت، آموزش، مالی و حتی دولت‌ها از فناوری‌های متن‌باز استفاده می‌کنند. برای مثال، سیستم عامل لینوکس و پایگاه داده MySQL نمونه‌هایی از فناوری‌های متن‌باز هستند که به طور گسترده در صنایع مختلف مورد استفاده قرار می‌گیرند. این فناوری‌ها نه تنها هزینه‌های توسعه را کاهش می‌دهند، بلکه انعطاف‌پذیری و قابلیت سفارشی‌سازی بالایی را نیز ارائه می‌کنند. با این حال، استفاده از فناوری‌های متن‌باز چالش‌هایی نیز به همراه دارد. یکی از مهم‌ترین این چالش‌ها، مدیریت امنیت است. اگرچه جامعه متن‌باز می‌تواند به شناسایی و رفع سریع‌تر آسیب‌پذیری‌ها کمک کند، اما برخی پروژه‌های متن‌باز ممکن است به اندازه کافی پشتیبانی نشوند یا به‌روزرسانی‌های منظم دریافت نکنند. بنابراین، سازمان‌ها باید در انتخاب و استفاده از این فناوری‌ها دقت لازم را به کار بگیرند.

هدف این کتاب این است که خوانندگان را با مفاهیم پایه‌ای متن‌باز آشنا کند و به آنها نشان دهد که چگونه می‌توانند این فناوری‌ها را در پروژه‌های واقعی به کار بگیرند. با ارائه مثال‌های عملی و استفاده از موارد واقعی، این کتاب به شرکت‌ها و دانشجویان کمک می‌کند تا نه تنها دانش نظری خود را افزایش دهند، بلکه مهارت‌های عملی لازم برای استفاده موثر از فناوری‌های متن‌باز را نیز کسب کنند.

تکنولوژی‌های متن‌باز (Open Source Technologies) به نرم‌افزارها، ابزارها یا سیستم‌هایی اشاره دارند که کد منبع (Source Code) آنها به صورت عمومی در دسترس است و کاربران می‌توانند آن را مشاهده، استفاده، تغییر و توزیع کنند. این مفهوم بر پایه همکاری جمعی، شفافیت و آزادی در استفاده از فناوری بنا شده است. در این پاسخ، به توضیح مفهوم تکنولوژی‌های متن‌باز، تاریخچه، ویژگی‌ها، مزایا، معایب و مثال‌هایی از آنها می‌پردازم و به سؤالات پاسخ می‌دهم که این تکنولوژی‌ها چه هستند. طبق دستورالعمل، کلمات انگلیسی را بولد کرده و ترجمه فارسی آنها را در پرانتز می‌آورم، و برای کلمات تکراری این کار را تکرار نمی‌کنم.

## تکنولوژی متن باز چیست؟

تکنولوژی **Open Source** (متن باز) به نرم‌افزاری گفته می‌شود که تحت مجوزهای خاصی (Licenses) منتشر می‌شود که به کاربران اجازه می‌دهد به **Source Code** (کد منبع) دسترسی داشته باشند و آن را مطابق نیاز خود تغییر دهند. برخلاف نرم‌افزارهای اختصاصی (Proprietary Software) که کدشان مخفی است و فقط شرکت سازنده می‌تواند آن را تغییر دهد، تکنولوژی‌های متن باز به جامعه‌ای از توسعه‌دهندگان (Developers) وابسته هستند که به صورت داوطلبانه یا حرفه‌ای در توسعه و بهبود آن مشارکت می‌کنند.

تکنولوژی **Open Source** (متن باز) به نوع خاصی از نرم‌افزارها اشاره دارد که تحت شرایط و مجوزهای ویژه‌ای منتشر می‌شوند و به کاربران این امکان را می‌دهند که به جنبه‌های اصلی و بنیادین آن نرم‌افزار دسترسی پیدا کنند. این مجوزها که به آن‌ها **Licenses** (مجوزها) گفته می‌شود، به افراد اجازه می‌دهند تا به **Source Code** (کد منبع) نرم‌افزار دست یابند، آن را مطالعه کنند، تغییر دهند و حتی بر اساس نیازهای خودشان آن را سفارشی‌سازی کنند. این ویژگی باعث می‌شود که کاربران نه تنها مصرف‌کننده صرف باشند، بلکه بتوانند به عنوان مشارکت‌کنندگان فعال در بهبود و توسعه نرم‌افزار نقش داشته باشند. در واقع، این نوع تکنولوژی به افراد این آزادی را می‌دهد که کنترل بیشتری بر ابزارهایی که استفاده می‌کنند داشته باشند و وابستگی کمتری به شرکت‌های بزرگ و انحصاری پیدا کنند.

در مقابل، نرم‌افزارهای **Proprietary Software** (نرم‌افزار اختصاصی) وجود دارند که رویکرد کاملاً متفاوتی را دنبال می‌کنند. در این نوع نرم‌افزارها، کد منبع به صورت مخفی نگه داشته می‌شود و فقط شرکت سازنده یا توسعه‌دهنده اصلی حق دسترسی و تغییر آن را دارد. این بدان معناست که کاربران عادی هیچ‌گونه کنترلی بر عملکرد داخلی این نرم‌افزارها ندارند و تنها می‌توانند از آن به همان شکلی که ارائه شده استفاده کنند. اگر مشکلی در نرم‌افزار پیش بیاید یا نیاز به ویژگی جدیدی احساس شود، کاربر مجبور است منتظر بماند تا شرکت سازنده خودش این تغییرات را اعمال کند، که گاهی ممکن است زمان‌بر یا حتی غیرممکن باشد. اما در تکنولوژی **Open Source**، این محدودیت‌ها وجود ندارد و هر کسی که دانش کافی داشته باشد می‌تواند دست به کار شود و تغییراتی را که لازم می‌داند اعمال کند.

یکی از جنبه‌های جذاب تکنولوژی متن باز، وابستگی آن به جامعه‌ای گسترده از **Developers** (توسعه‌دهندگان) است. این افراد، که ممکن است به صورت داوطلبانه یا حرفه‌ای در این پروژه‌ها فعالیت کنند، نقش اساسی در رشد و بهبود این نرم‌افزارها دارند. برخی از توسعه‌دهندگان به دلیل علاقه شخصی و عشق به فناوری در این پروژه‌ها شرکت می‌کنند، در حالی که برخی دیگر ممکن است به عنوان بخشی از شغل خود یا برای کسب تجربه و شهرت حرفه‌ای وارد این حوزه شوند. این همکاری جمعی باعث می‌شود که نرم‌افزارهای متن باز به سرعت پیشرفت کنند، زیرا ایده‌ها و مهارت‌های متنوعی از سراسر جهان در آن‌ها ترکیب می‌شود. به عنوان مثال، پروژه‌هایی مثل سیستم‌عامل **Linux** (لینوکس) یا مرورگر **Firefox** (فایرفاکس) نمونه‌های موفق از این همکاری هستند که میلیون‌ها نفر در سراسر جهان از آن‌ها استفاده می‌کنند.

یکی از مزیت‌های بزرگ تکنولوژی **Open Source**، شفافیت آن است. از آنجا که کد منبع در دسترس همه قرار دارد، کاربران می‌توانند مطمئن شوند که نرم‌افزار دقیقاً همان کاری را انجام می‌دهد که ادعا می‌کند و هیچ‌گونه کد مخرب یا جاسوسی در آن پنهان نشده است. این موضوع به‌ویژه در دنیای امروز که نگرانی‌های مربوط به حریم خصوصی و امنیت داده‌ها روز به روز بیشتر می‌شود، اهمیت زیادی دارد. برای مثال، شرکت‌هایی که از نرم‌افزارهای اختصاصی استفاده می‌کنند، ممکن است ندانند که آیا اطلاعات حساس آن‌ها در پشت پرده جمع‌آوری می‌شود یا نه. اما در نرم‌افزارهای متن باز، هر خط کد قابل بررسی است و این اعتماد بیشتری را به کاربران می‌دهد.

علاوه بر این، تکنولوژی متن باز معمولاً هزینه کمتری نسبت به نرم‌افزارهای اختصاصی دارد. بسیاری از پروژه‌های **Open Source** به صورت رایگان در دسترس هستند و کاربران نیازی به پرداخت هزینه‌های سنگین برای خرید مجوز یا اشتراک ندارند. البته این به این معنا نیست که استفاده از آن‌ها همیشه کاملاً بدون هزینه باشد؛ گاهی اوقات ممکن است برای پشتیبانی فنی، آموزش یا سفارشی‌سازی نیاز به صرف هزینه باشد. با این حال، این هزینه‌ها معمولاً بسیار کمتر از نرم‌افزارهای تجاری است و انعطاف‌پذیری بیشتری به کاربران می‌دهد. شرکت‌ها و سازمان‌هایی که بودجه محدودی دارند، اغلب به سمت این نوع تکنولوژی گرایش پیدا می‌کنند تا بتوانند نیازهای خود را با کمترین هزینه برآورده کنند.

یکی دیگر از ویژگی‌های برجسته تکنولوژی **Open Source**، قابلیت همکاری و انعطاف‌پذیری آن است. از آنجا که کاربران می‌توانند کد منبع را تغییر دهند، این نرم‌افزارها را می‌توان به راحتی با سیستم‌های دیگر سازگار کرد یا برای کاربردهای خاص بازطراحی نمود. این موضوع برای کسب‌وکارهایی که نیاز به راه‌حل‌های سفارشی دارند بسیار مفید است. به عنوان مثال، یک شرکت ممکن است بخواهد نرم‌افزاری را با فرآیندهای داخلی خود هماهنگ کند یا ویژگی‌هایی را به آن اضافه کند که در نسخه اصلی وجود ندارد. در نرم‌افزارهای اختصاصی، چنین تغییراتی معمولاً غیرممکن است یا نیاز به مذاکره طولانی با شرکت سازنده دارد، اما در دنیای متن باز، این آزادی به کاربران داده شده است.

با این حال، استفاده از تکنولوژی **Open Source** بدون چالش نیست. یکی از مشکلات اصلی این است که گاهی اوقات پشتیبانی رسمی برای این نرم‌افزارها وجود ندارد. برخلاف نرم‌افزارهای اختصاصی که معمولاً توسط شرکت سازنده پشتیبانی می‌شوند، در پروژه‌های متن باز اغلب باید به جامعه کاربران یا توسعه‌دهندگان متکی بود. این موضوع می‌تواند برای کاربرانی که تجربه فنی کمی دارند مشکل‌ساز شود، زیرا ممکن است در حل مشکلات خود به تنهایی ناتوان باشند. با این وجود، بسیاری از پروژه‌های بزرگ متن باز، مانند **Apache** (آپاچی) یا **MySQL** (مای‌اس‌کیوال)، انجمن‌های فعالی دارند که به کاربران کمک می‌کنند تا مشکلات خود را برطرف کنند.

از سوی دیگر، یادگیری و استفاده از نرم‌افزارهای متن باز گاهی نیاز به دانش فنی بیشتری دارد. برای مثال، کار با **Source Code** و تغییر آن نیازمند مهارت‌های برنامه‌نویسی است که همه کاربران آن را ندارند. این موضوع ممکن است باعث شود که برخی افراد یا شرکت‌ها ترجیح دهند به سراغ نرم‌افزارهای آماده و ساده‌تر بروند، حتی اگر هزینه بیشتری داشته باشند. اما برای کسانی که مایل به سرمایه‌گذاری در یادگیری هستند، این تکنولوژی می‌تواند دریچه‌ای به سوی امکانات بی‌پایان باشد.

تأثیر تکنولوژی **Open Source** تنها به حوزه نرم افزار محدود نمی شود. این مفهوم به تدریج به سایر زمینه ها مانند سخت افزار، آموزش و حتی علم نیز گسترش یافته است. به عنوان مثال، پروژه هایی مثل **Arduino** (آردوینو) در حوزه سخت افزار متن باز، به افراد اجازه می دهند تا دستگاه های الکترونیکی خود را بسازند و تغییر دهند. در آموزش نیز، منابع آموزشی باز مانند کتاب ها و دوره های آنلاین رایگان، دانش را برای همه قابل دسترس کرده اند. این رویکرد مشارکتی و باز، فرهنگ نوآوری را ترویج می دهد و به افراد اجازه می دهد که به جای مصرف کننده بودن، خالق باشند. تکنولوژی **Open Source** نشان دهنده یک فلسفه بزرگ تر است: این که دانش و فناوری باید در خدمت همه باشد، نه فقط عده ای خاص. این ایدئولوژی باعث شده که میلیون ها نفر در سراسر جهان به آن جذب شوند و در پروژه های مختلف مشارکت کنند. از شرکت های بزرگ فناوری گرفته تا افراد عادی، همه می توانند سهمی در این حرکت داشته باشند. این تکنولوژی نه تنها ابزارهایی قدرتمند در اختیار ما قرار داده، بلکه راهی برای همکاری، خلاقیت و پیشرفت جمعی را نیز نشان داده است. در دنیایی که هر روز بیشتر به فناوری وابسته می شویم، شاید این رویکرد بتواند به ما کمک کند تا آینده ای بهتر و عادلانه تر بسازیم.

این تکنولوژی ها معمولاً با مجوزهایی مثل **GNU General Public License (GPL)** - مجوز عمومی گنو، **MIT License** (مجوز ام آی تی) یا **Apache License** (مجوز آپاچی) عرضه می شوند که شرایط استفاده و توزیع را مشخص می کنند. هدف اصلی این رویکرد، ایجاد نرم افزاری است که آزاد (Free)، قابل اعتماد و قابل انطباق با نیازهای مختلف باشد.

### تاریخچه تکنولوژی های متن باز

جنبش **Open Source** ریشه در دهه های ۱۹۸۰ و ۱۹۹۰ دارد، زمانی که ریچارد استالمن (Richard Stallman) جنبش نرم افزار آزاد (Free Software Movement) را آغاز کرد. او در سال ۱۹۸۳ پروژه **GNU** (گنو) را راه اندازی کرد تا سیستمی کاملاً آزاد بسازد که جایگزین سیستم های اختصاصی شود. در سال ۱۹۹۱، لینوس توروالدز (Linus Torvalds) سیستم عامل **Linux** (لینوکس) را معرفی کرد که بر پایه ایده های **GNU** ساخته شد و به یکی از موفق ترین پروژه های متن باز تبدیل شد. در سال ۱۹۹۸، اصطلاح **Open Source** توسط افرادی مثل اریک ریموند (Eric Raymond) و بروس پرنز (Bruce Perens) به طور رسمی مطرح شد تا این مفهوم را از نظر تجاری و کاربردی جذاب تر کند. از آن زمان، شرکت های بزرگ مثل **IBM** (آی بی ام)، **Google** (گوگل) و **Microsoft** (مایکروسافت) نیز به این جنبش پیوستند و پروژه های متن باز را حمایت کردند.

## فصل ۱

### منبع باز

#### چگونه به اینجا رسیدیم

در دنیای امروز، فناوری **منبع باز (Open Source)** به عنوان ستون فقرات نوآوری و مدرنیزاسیون دیجیتال در نظر گرفته می‌شود. اما سوال اینجاست: **متن باز (Open Source)** به چه معناست؟ در ساده‌ترین تعریف، **متن باز** به نرم‌افزاری اشاره دارد که هر کسی می‌تواند آن را تغییر داده و به اشتراک بگذارد، زیرا طراحی آن به راحتی برای عموم قابل دسترسی است. این ویژگی باعث شده است که **متن باز** به یکی از پایه‌های اصلی توسعه فناوری در جهان تبدیل شود. فناوری‌های **منبع باز** به افزایش سرعت نوآوری کمک می‌کنند، زیرا باز بودن آنها امکان همکاری گسترده را فراهم می‌آورد. این همکاری‌ها منجر به تبادل **ایده‌های نوآورانه (Innovative Ideas)** در جوامع مختلف می‌شود. به عنوان مثال، فرض کنید یک کاربر نهایی یا یک توسعه‌دهنده، یک فناوری یا یک پروژه خاص را جذاب می‌بیند. در این صورت، آنها می‌توانند نرم‌افزار را دانلود کرده و آن را امتحان کنند. آنها می‌توانند ببینند که این نرم‌افزار چه کاری انجام می‌دهد و آیا نیازهای آنها را برآورده می‌کند یا خیر. اگر این نرم‌افزار نیازهای آنها را برآورده کرد، می‌توانند انتخاب کنند که به پروژه **منبع باز** بپیوندند و در آن مشارکت کنند. حتی می‌توانند از آن به عنوان پایه یا اساس برای ایجاد چیزی جدید استفاده کنند. امروزه، میلیون‌ها نفر در سراسر جهان از فناوری‌های **منبع باز** استفاده می‌کنند و این نشان‌دهنده گستردگی و محبوبیت این فناوری‌ها است. یکی از نمونه‌های برجسته فناوری **منبع باز**، سیستم‌عامل **اندروید (Android)** است که توسط **گوگل (Google)** راه‌اندازی شده است. **اندروید** یک سیستم‌عامل **متن باز** است که در ابتدا برای دستگاه‌های تلفن همراه ساخته شد. این سیستم‌عامل به دلیل انعطاف‌پذیری و قابلیت‌های گسترده‌اش، به یکی از محبوب‌ترین سیستم‌عامل‌های موبایل در جهان تبدیل شده است. سایر نمونه‌های نرم‌افزار **منبع باز** شامل **وب سرور آپاچی (Apache Web Server)**، **مرورگر موزیلا فایرفاکس (Mozilla Firefox)**، **بrower، لیبره آفیس (LibreOffice)** و **سیستم‌عامل لینوکس (Linux Operating System)** است. هر یک از این نرم‌افزارها نقش مهمی در توسعه فناوری و ارائه راه‌حل‌های نوآورانه ایفا کرده‌اند.

#### مزایای فناوری‌های منبع باز

یکی از مهم‌ترین مزایای فناوری‌های **منبع باز**، شفافیت آنها است. از آنجایی که کدهای منبع به صورت عمومی در دسترس هستند، هر کسی می‌تواند آنها را بررسی کند و در صورت وجود اشکالات یا آسیب‌پذیری‌ها، آنها را گزارش دهد یا حتی خودش آنها را برطرف کند. این ویژگی باعث افزایش امنیت و کیفیت نرم‌افزارهای **منبع باز** می‌شود. علاوه بر این، **منبع باز** هزینه‌های توسعه را کاهش می‌دهد، زیرا سازمان‌ها و افراد می‌توانند از نرم‌افزارهای موجود استفاده کنند و بدون نیاز به شروع از صفر، پروژه‌های خود را پیش ببرند.

- **FLOSS** مخفف **Free/Libre and Open Source Software** (نرم افزار متن باز رایگان/آزاد) است. این اصطلاح نیز مشابه **FOSS** است، اما تأکید بیشتری بر مفهوم "آزادی" (**Libre**) دارد. کلمه **Libre** از زبان لاتین گرفته شده و به معنای "آزادی کامل" یا "آزادی" است. این اصطلاح برای جلوگیری از ابهام کلمه **Free** در زبان انگلیسی (که هم به معنای "رایگان" و هم به معنای "آزاد" است) استفاده می شود. بنیاد نرم افزار آزاد (**Free Software Foundation** یا **FSF**) از استفاده از اصطلاح **FLOSS** حمایت می کند.

### ابهام کلمه Free در زبان انگلیسی

در فرهنگ لغت دانشگاهی **Merriam-Webster**، دو تعریف اصلی برای کلمه **Free** وجود دارد:

۱. هزینه یا هزینه نکردن چیزی) رایگان بودن
۲. لذت بردن از آزادی شخصی: تحت کنترل یا تسلط دیگری نبودن) آزاد بودن

با توجه به این ابهام، اصطلاح **Libre** برای تأکید بر مفهوم "آزادی" انتخاب شده است. این اصطلاح به این معناست که کاربران نه تنها می توانند نرم افزار را به صورت رایگان استفاده کنند، بلکه آزادی تغییر و توزیع مجدد آن را نیز دارند. این مفهوم توسط **ریچارد استالمن (Richard Stallman)**، بنیان گذار جنبش نرم افزار آزاد، مطرح شده است. او تأکید می کند که **Free** در اینجا به معنای "آزادی" است، نه لزوماً "رایگان بودن".

### اصطلاحات رایج دیگر

علاوه بر **FOSS** و **FLOSS**، اصطلاحات دیگری نیز در دنیای نرم افزار وجود دارند که ممکن است با نرم افزار متن باز اشتباه گرفته شوند. دو نمونه از این اصطلاحات، **نرم افزار رایگان (Freeware)** و **اشتراک افزار (Shareware)** هستند.

#### ۱. نرم افزار رایگان: (Freeware)

- نرم افزار رایگان به نرم افزاری گفته می شود که کاربران می توانند آن را بدون پرداخت هیچ هزینه ای دریافت و استفاده کنند. با این حال، این نرم افزارها لزوماً متن باز نیستند و کاربران آزادی تغییر یا توزیع مجدد آنها را ندارند.
- یکی از جنبه های کلیدی نرم افزار رایگان این است که این نرم افزارها معمولاً منبع بسته (**Closed Source**) هستند، به این معنا که کد منبع آنها در دسترس عموم قرار نمی گیرد.
- یک نمونه معروف از نرم افزار رایگان، **Adobe Acrobat Reader** است. این نرم افزار به کاربران اجازه می دهد تا فایل های **PDF** را مشاهده کنند، اما کاربران نمی توانند کد منبع آن را تغییر دهند یا آن را به صورت مجدد توزیع کنند.

#### ۲. اشتراک افزار: (Shareware)



### فرمت: PDF: استاندارد باز و جهانی

فرمت PDF که مخفف **Portable Document Format** (قالب سند قابل حمل) است، یک استاندارد باز است که توسط شرکت **Adobe** توسعه داده شده است. این فرمت به طور **free** (رایگان) در دسترس همه قرار دارد و تقریباً در همه دستگاه‌ها و سیستم‌عامل‌ها قابل مشاهده است. برخی از ویژگی‌های کلیدی فرمت PDF عبارتند از:

۱. **سازگاری گسترده:** فایل‌های PDF تقریباً در هر دستگاهی، از جمله کامپیوترهای شخصی، لپ‌تاپ‌ها، تلفن‌های هوشمند و تبلت‌ها، قابل باز کردن هستند. این سازگاری به دلیل وجود نرم‌افزارهای رایگان و متن‌باز مانند **Adobe Acrobat Reader** و سایر برنامه‌های PDF خوان است.
۲. **حفظ قالب‌بندی:** یکی از بزرگ‌ترین مزایای PDF این است که قالب‌بندی سند (مانند فونت‌ها، تصاویر و چیدمان) در همه دستگاه‌ها به‌طور یکسان نمایش داده می‌شود. این ویژگی باعث می‌شود PDF برای ارسال اسناد رسمی، گزارش‌ها و مدارک مهم ایده‌آل باشد.
۳. **امنیت PDF:** امکان اضافه کردن رمز عبور، محدودیت‌های ویرایش و امضای دیجیتال را فراهم می‌کند، که آن را به یک گزینه امن برای اسناد حساس تبدیل می‌کند.
۴. **اندازه فایل کوچک:** فایل‌های PDF معمولاً حجم کمتری نسبت به فایل‌های Word دارند، که این موضوع باعث می‌شود ارسال و دانلود آنها سریع‌تر و آسان‌تر باشد.

### فرمت: DOCX: استاندارد باز مبتنی بر XML

فرمت **.docx** که توسط **Microsoft Word** استفاده می‌شود، بر اساس یک استاندارد باز به نام **Office Open XML** (اکس‌ام‌ال دفتر باز) توسعه داده شده است. این فرمت نیز به‌طور گسترده‌ای پشتیبانی می‌شود و تقریباً در همه دستگاه‌ها و سیستم‌عامل‌ها قابل مشاهده است. برخی از ویژگی‌های کلیدی فرمت DOCX عبارتند از:

۱. **سازگاری گسترده:** مانند PDF، فایل‌های DOCX نیز در اکثر دستگاه‌ها و سیستم‌عامل‌ها قابل باز کردن هستند. برنامه‌هایی مانند **Microsoft Word**، **Google Docs**، **LibreOffice** و **Apple Pages** از این فرمت پشتیبانی می‌کنند.
۲. **قابلیت ویرایش:** برخلاف PDF، فایل‌های DOCX به راحتی قابل ویرایش هستند. این ویژگی باعث می‌شود که این فرمت برای اسنادی که نیاز به تغییرات مکرر دارند، مانند پیش‌نویس‌ها یا پروژه‌های گروهی، مناسب باشد.

آن ارائه دهد. اگرچه پیاده‌سازی سیستم‌عامل ممکن است در پلتفرم‌های مختلف متفاوت باشد، اما عملکرد ارائه‌شده به برنامه‌ها یکسان است. به همین دلیل، یک برنامه مانند مرورگر **فایرفاکس (Firefox)** می‌تواند روی سیستم‌عامل‌های مختلف اجرا شود، به شرطی که برای هر سیستم‌عامل به‌طور جداگانه ساخته شده باشد. این توضیح می‌دهد که چرا فایل‌های قابل دانلود جداگانه‌ای برای **فایرفاکس** در سیستم‌عامل‌های مختلف مانند **ویندوز (Windows)**، **مک‌اواس (macOS)** و **لینوکس (Linux)** وجود دارد.

### فایرفاکس

به‌عنوان یک نرم‌افزار متن‌باز (**Open Source Software**)، تمام رابط‌های مختلف سیستم‌عامل‌ها را خلاصه می‌کند و به شما امکان می‌دهد تجربه یکسانی در استفاده از مرورگر داشته باشید. این کاری است که با یک نرم‌افزار متن‌بسته (**Closed Source Software**) به‌سختی می‌توان انجام داد، زیرا چنین نرم‌افزاری به‌احتمال زیاد برای یک سیستم‌عامل خاص، معمولاً متعلق به شرکت مادر توسعه‌دهنده، ساخته شده است. علاوه بر این، جامعه منبع باز (**Open Source Community**) تضمین می‌کند که نرم‌افزار حداکثر سازگاری را با چندین سیستم‌عامل داشته باشد. این جامعه با مشارکت فعال توسعه‌دهندگان از سراسر جهان، به بهبود مستمر نرم‌افزار و تطبیق آن با نیازهای کاربران مختلف کمک می‌کند.

### استانداردهای باز و نوآوری

استانداردهای باز نه‌تنها همکاری بین سیستم‌های مختلف را تسهیل می‌کنند، بلکه به نوآوری نیز کمک می‌کنند. زمانی که یک استاندارد باز وجود دارد، توسعه‌دهندگان می‌توانند بر روی بهبود و توسعه محصولات خود تمرکز کنند، بدون اینکه نگران مسائل مربوط به سازگاری و تعامل‌پذیری باشند. به‌عنوان مثال، استانداردهای باز در حوزه شبکه‌های کامپیوتری، مانند **TCP/IP (Transmission Control Protocol/Internet Protocol)**، امکان ارتباط بین دستگاه‌های مختلف در سراسر جهان را فراهم کرده‌اند. این استانداردها به‌عنوان پایه‌ای برای اینترنت عمل می‌کنند و بدون آن‌ها، اینترنت به‌شکل امروزی وجود نداشت.

### چالش‌های استانداردهای باز

با وجود مزایای فراوان، استانداردهای باز با چالش‌هایی نیز مواجه هستند. یکی از این چالش‌ها، تضمین مشارکت عادلانه همه ذینفعان در فرآیند توسعه استانداردها است. در برخی موارد، شرکت‌های بزرگ ممکن است سعی کنند استانداردها را به نفع خود تغییر دهند، که این امر می‌تواند به ایجاد استانداردهای ناعادلانه و غیرمنصفانه منجر شود. برای جلوگیری از این مسئله، سازمان‌های توسعه‌دهنده استانداردها باید مکانیسم‌هایی برای تضمین مشارکت عادلانه همه ذینفعان ایجاد کنند.

### نقش جامعه منبع باز در استانداردسازی

جامعه منبع باز نقش کلیدی در توسعه و ترویج استانداردهای باز ایفا می‌کند. پروژه‌های منبع باز اغلب به‌عنوان آزمایشگاهی برای ایده‌های جدید عمل می‌کنند و می‌توانند به ایجاد استانداردهای جدید کمک کنند. به‌عنوان مثال، پروژه‌های منبع باز در حوزه‌های مختلف فناوری، مانند **لینوکس (Linux)** و **آپاچی (Apache)**، به ایجاد

از **OpenStack** و **OpenvSwitch** می‌تواند به مدیریت بهتر منابع در یک دیتاستر کمک کند، در حالی که **ONAP** و **OpenDaylight** می‌توانند هماهنگی بیشتری در کنترل شبکه ایجاد کنند. مزیت بزرگ این پلتفرم‌ها این است که منبع باز بودن آن‌ها به توسعه‌دهندگان اجازه می‌دهد تا کدهای موجود را بررسی کرده و بر اساس نیازهای خود تغییراتی در آن‌ها اعمال کنند. این ویژگی باعث شده که جامعه‌ای فعال از برنامه‌نویسان در سراسر جهان به بهبود مداوم این ابزارها کمک کنند. علاوه بر این، مستندات گسترده‌ای که برای این پروژه‌ها ارائه می‌شود، یادگیری و استفاده از آن‌ها را برای کاربران جدید آسان‌تر می‌کند. برای مثال، اگر یک دانشجوی فناوری اطلاعات بخواهد با مفاهیم شبکه‌های مدرن آشنا شود، می‌تواند به راحتی به مستندات **OpenDaylight** دسترسی پیدا کند و با استفاده از آن یک پروژه عملی را پیاده‌سازی کند. پیاده‌سازی فناوری ارتباطات در جوامع منبع باز نشان‌دهنده یک حرکت رو به جلو در جهت دموکراتیزه کردن دانش و فناوری است. ابزارهایی مانند **ONAP**، **OpenDaylight**، **OpenvSwitch** و **OpenStack** نه تنها به بهبود زیرساخت‌های شبکه کمک می‌کنند، بلکه فرصتی را برای افراد و سازمان‌ها فراهم می‌کنند تا بدون وابستگی به شرکت‌های بزرگ تجاری، به ابزارهای قدرتمند دسترسی داشته باشند. بنیاد **لینوکس** با ادامه حمایت از این پروژه‌ها، نقشی کلیدی در آینده فناوری ایفا می‌کند و به توسعه‌دهندگان و کاربران اطمینان می‌دهد که می‌توانند به ابزارهایی قابل اعتماد و به روز دسترسی داشته باشند. این رویکرد منبع باز، نوآوری را تسریع کرده و به همه اجازه می‌دهد تا در ساخت آینده‌ای بهتر مشارکت کنند.

صنعت خودروسازی در دنیای معاصر دستخوش تغییرات عمیقی شده است و دیگر شبیه به چیزی که چند سال پیش بود نیست. با پیشرفت سریع تکنولوژی، وسایل نقلیه روز به روز هوشمندتر می‌شوند و از حالت صرفاً مکانیکی خارج شده‌اند. این روزها، خودروها به دستگاه‌های پیشرفته‌ای تبدیل شده‌اند که مجهز به سیستم‌های پیچیده و هوشمند هستند و عملکرد آن‌ها دیگر تنها به قطعات مکانیکی محدود نمی‌شود. در واقع، وسایل نقلیه امروزی با استفاده از **مدارهای الکترونیکی قابل برنامه‌ریزی (Programmable Electronic Circuits)** و میلیون‌ها خط **کد (Code)** نرم‌افزاری، قابلیت‌های جدیدی پیدا کرده‌اند که پیش‌تر تصور آن‌ها دشوار بود. این پیشرفت‌ها به خودروها امکان می‌دهد تا وظایف متنوعی مانند کنترل موتور، ارائه اطلاعات سرگرمی، برقراری ارتباطات و حتی ویژگی‌های رانندگی خودکار یا **خودران (Autonomous)** را به انجام برسانند. این تغییرات، استانداردهای حاکم بر صنعت خودرو را نیز تحت تأثیر قرار داده و از تمرکز صرف بر کیفیت و ایمنی، به سمت نیازهای خاص فناوری در حوزه‌هایی مثل **ناوبری (Navigation)**، **ارتباطات (Communication)** و سایر جنبه‌های پیشرفته سوق داده است.

در گذشته، استانداردهای خودروسازی بیشتر بر جنبه‌های فیزیکی و ایمنی متمرکز بودند، اما حالا با ورود فناوری‌های نوین، این استانداردها گسترده‌تر شده‌اند. برای مثال، امروزه استانداردهایی برای اطمینان از عملکرد صحیح سیستم‌های الکترونیکی و نرم‌افزاری در خودروها تدوین شده است. این استانداردها نه تنها باید ایمنی سرنشینان را

در حوزه سیستم‌عامل‌ها نیز، **FreeBSD** نمونه برجسته‌ای از استفاده از مجوز **BSD** است. **FreeBSD** یک سیستم‌عامل **open source** است که بر پایه یونیکس توسعه یافته و به دلیل پایداری و امنیت بالا شناخته می‌شود. این سیستم‌عامل در سرورها، سیستم‌های نهفته و حتی برخی از محصولات تجاری مانند کنسول‌های بازی به کار گرفته شده است. شرکت‌هایی مانند نتفلیکس و سونی از **FreeBSD** در زیرساخت‌های خود استفاده می‌کنند، که نشان‌دهنده قابلیت اطمینان بالای آن است. مجوز **BSD** به توسعه‌دهندگان این امکان را داده که این سیستم‌عامل را برای کاربردهای مختلف سفارشی‌سازی کنند و در عین حال از کد منبع آن در محصولات خود بهره ببرند.

یکی دیگر از پروژه‌های قابل توجه، نرم‌افزار **Bionic** است که به عنوان پیاده‌سازی استاندارد **C library** (کتابخانه سی) برای سیستم‌عامل اندروید عمل می‌کند. **Bionic** که تحت مجوز **BSD** منتشر شده، نقش مهمی در عملکرد روان و بهینه اندروید ایفا می‌کند. این کتابخانه به توسعه‌دهندگان اندروید کمک می‌کند تا برنامه‌هایی با کارایی بالا بسازند که با سخت‌افزارهای مختلف سازگار باشند. استفاده از مجوز **BSD** در این پروژه تضمین می‌کند که گوگل و دیگر شرکت‌ها می‌توانند تغییرات مورد نیاز خود را در این کتابخانه اعمال کنند، بدون اینکه ملزم به اشتراک‌گذاری این تغییرات با عموم باشند.

مزیت اصلی مجوز **BSD** در مقایسه با مجوزهای دیگر، مانند **GPL** (مجوز عمومی گنو)، این است که به کاربران آزادی بیشتری می‌دهد. برخلاف **GPL** که توسعه‌دهندگان را ملزم می‌کند تغییرات خود را منتشر کنند، **BSD** این اجبار را ندارد. این ویژگی باعث شده که شرکت‌های تجاری با اطمینان بیشتری از پروژه‌های تحت این مجوز استفاده کنند، زیرا می‌توانند کد را خصوصی نگه دارند و در عین حال از مزایای جامعه **open source** بهره‌مند شوند. این انعطاف‌پذیری به پروژه‌هایی مانند **Go**، **Nginx**، **FreeBSD** و **Bionic** اجازه داده که هم در میان توسعه‌دهندگان فردی و هم در میان شرکت‌های بزرگ محبوب شوند.

مجوز **BSD** با تاریخچه غنی و سادگی خود، نقش مهمی در توسعه نرم‌افزارهای مدرن ایفا کرده است. از زبان‌های برنامه‌نویسی گرفته تا وب‌سرورها و سیستم‌عامل‌ها، این مجوز به پروژه‌ها کمک کرده که هم نوآورانه باشند و هم کاربردی. با توجه به اینکه فناوری همچنان در حال پیشرفت است، انتظار می‌رود که پروژه‌های بیشتری تحت این مجوز توسعه یابند و به تکامل دنیای دیجیتال کمک کنند. موفقیت این پروژه‌ها نشان‌دهنده قدرت همکاری و انعطاف‌پذیری در اکوسیستم **open source** است که مجوز **BSD** به خوبی آن را نمایندگی می‌کند.

### مجوز عمومی عمومی گنو

مجوز ( **GNU General Public License (GPL)** مجوز عمومی عمومی گنو)، که به سادگی به عنوان **GPL** شناخته می‌شود، مجموعه‌ای از مجوزهای نرم‌افزاری است که توسط ریچارد استالمن، بنیان‌گذار جنبش **free software** (نرم‌افزار آزاد)، ایجاد شده است. این مجوزها با هدف تضمین آزادی کاربران در استفاده، تغییر و توزیع نرم‌افزارها طراحی شده‌اند و در دسته مجوزهای **reciprocal** (متقابل) قرار می‌گیرند. این بدان معناست که هر نرم‌افزاری که از کد تحت **GPL** استفاده کند، باید خود نیز تحت همین مجوز منتشر شود تا اصول

## Linux kernel (هسته لینوکس)

یکی از معروفترین پروژه‌هایی که از **GPL** بهره می‌برد، **Linux kernel** (هسته لینوکس) است. این هسته، که توسط لینوس توروالدس در سال ۱۹۹۱ توسعه یافت، تحت نسخه دوم این مجوز یعنی **GPLv۲** منتشر شده است. **Linux kernel** به عنوان قلب بسیاری از سیستم‌عامل‌ها، از جمله سیستم‌عامل **GNU/Linux** (گنو/لینوکس)، عمل می‌کند و پایه‌ای برای توزیع‌های متنوعی مانند اوبونتو، دبیان و فدورا فراهم کرده است. این پروژه به دلیل استفاده از **GPL** توانسته به یک تلاش جهانی تبدیل شود، جایی که توسعه‌دهندگان از سراسر جهان کدهای خود را به اشتراک می‌گذارند و بهبود می‌بخشند. الزام **GPL** به ارائه کد منبع به همراه هر نسخه توزیع شده، تضمین کرده که این هسته آزاد باقی بماند و هیچ شرکتی نتواند آن را به طور کامل به یک محصول اختصاصی تبدیل کند. این ویژگی باعث شده که **Linux** در سرورها، دستگاه‌های نهفته، گوشی‌های هوشمند و حتی ابررایانه‌ها به طور گسترده استفاده شود. برای مثال، سیستم‌عامل اندروید، که توسط گوگل توسعه یافته، از نسخه تغییرشده‌ای از **Linux kernel** بهره می‌برد و این نشان‌دهنده انعطاف‌پذیری و قدرت این پروژه تحت مجوز **GPL** است.

## MySQL و MariaDB

در حوزه پایگاه‌های داده، دو پروژه برجسته به نام‌های **MySQL** و **MariaDB** نیز تحت **GPL** منتشر شده‌اند. نسخه جامعه **MySQL**، که یک سیستم مدیریت پایگاه داده رابطه‌ای محبوب است، از **GPLv۲** استفاده می‌کند. این پایگاه داده به دلیل سرعت، قابلیت اطمینان و سهولت استفاده، در بسیاری از برنامه‌های وب مانند وردپرس و جوملا به کار گرفته شده است. اما پس از خرید **MySQL** توسط شرکت اوراکل، نگرانی‌هایی درباره مسیر آینده آن ایجاد شد. در نتیجه، جامعه توسعه‌دهندگان یک **fork** (انشعاب) از آن به نام **MariaDB** را ایجاد کردند که آن هم تحت **GPLv۲** منتشر شده است. **MariaDB** نه تنها جایگزینی برای **MySQL** ارائه داد، بلکه ویژگی‌های جدیدی مانند عملکرد بهتر و پشتیبانی از موتورهای ذخیره‌سازی متنوع را نیز به ارمغان آورد. استفاده از **GPL** در این پروژه‌ها به کاربران اطمینان داده که کد منبع همیشه در دسترس خواهد بود و می‌توانند آن را بر اساس نیازهای خود تغییر دهند، که این یکی از دلایل محبوبیت گسترده آن‌ها در میان توسعه‌دهندگان وب و شرکت‌های فناوری است.

## Drupal

یکی دیگر از پروژه‌های قابل توجه، **Drupal** است که یک سیستم مدیریت محتوای **open source** (متن‌باز) است و تحت **GPLv۲** یا نسخه‌های جدیدتر مجوز دارد. **Drupal** به دلیل انعطاف‌پذیری و قابلیت‌های پیشرفته‌اش، برای ساخت وبسایت‌های پیچیده و پویا مورد استفاده قرار می‌گیرد. از وبسایت‌های دولتی گرفته تا پلتفرم‌های تجاری، این سیستم به توسعه‌دهندگان اجازه می‌دهد ماژول‌های سفارشی بسازند و آن را با نیازهای خاص خود تطبیق دهند. الزام **GPL** به اشتراک‌گذاری کد منبع باعث شده که جامعه بزرگی از توسعه‌دهندگان به بهبود مداوم **Drupal** کمک کنند و هزاران ماژول و تم رایگان در دسترس کاربران قرار گیرد. این همکاری

و تغییرات در کد منبع ارتباط دارد. شرکت ها می توانند با مشارکت در پروژه های **open source**، تغییراتی در کد ایجاد کنند که نه تنها نیازهای خاص خودشان را برآورده کند، بلکه به آن ها اجازه دهد محصولی متمایز ارائه دهند. این تغییرات می توانند شامل افزودن قابلیت های جدید، بهبود رابط کاربری، یا حتی بهینه سازی هایی باشند که تجربه کاربر را ارتقا می دهند. اما این فرآیند به انتخاب مجوز مناسب بستگی دارد، زیرا هر مجوز شرایط خاص خود را برای استفاده، اصلاح و توزیع کد تعیین می کند.

## اندروید

یکی از بهترین نمونه ها برای توضیح این مفهوم، سیستم عامل **Android** (اندروید) است که به عنوان یک نرم افزار **open source** شناخته می شود و تحت مجوزهای مختلفی منتشر شده است **Android**. توسط بسیاری از تولیدکنندگان گوشی های هوشمند مانند سامسونگ، شیائومی و هواوی استفاده می شود و به لطف انعطاف پذیری اش، به پرستفاده ترین سیستم عامل موبایل در جهان تبدیل شده است. مجوز اصلی و ترجیحی **Android**، **Apache License ۲,۰** (مجوز آپاچی نسخه ۲,۰) است که یک مجوز **permissive** (مجاز) محسوب می شود و به توسعه دهندگان آزادی زیادی برای تغییر و استفاده تجاری از کد می دهد، بدون اینکه آن ها را ملزم به اشتراک گذاری تغییراتشان کند. با این حال، **Android** تنها به این مجوز محدود نمی شود؛ این سیستم عامل شامل **Linux kernel** (هسته لینوکس) است که تحت **GPL** (مجوز عمومی گنو) منتشر شده و همچنین اجزای دیگری با مجوزهای متنوع مانند **LGPL** (مجوز عمومی عمومی کوچک گنو) دارد. این وضعیت، که به **multi-licensing** (صدور مجوز چندگانه) معروف است، در دنیای **open source** بیشتر یک هنجار است تا یک استثنا و نشان دهنده پیچیدگی هایی است که شرکت ها هنگام کار با پروژه های متن باز با آن مواجه می شوند.

## Multi-licensing

**Multi-licensing** به این معناست که یک پروژه واحد می تواند بخش هایی با مجوزهای مختلف داشته باشد و هر بخش شرایط خاص خود را برای استفاده و توزیع تعیین کند. در مورد **Android**، این تنوع مجوزها به تولیدکنندگان اجازه می دهد که بسته به بخشی از کد که تغییر می دهند، استراتژی های متفاوتی را اتخاذ کنند. برای مثال، تغییراتی که در بخش های تحت **Apache ۲,۰** اعمال می شوند، می توانند به صورت **proprietary** (اختصاصی) باقی بمانند، در حالی که تغییراتی در **Linux kernel** تحت **GPL** باید به جامعه باز گردانده شوند و کد منبع آن ها در دسترس قرار گیرد. این تفاوت ها به شرکت ها امکان می دهد که تصمیم بگیرند کدام بخش از محصولشان را باز نگه دارند و کدام بخش را به عنوان یک مزیت رقابتی خصوصی حفظ کنند. به همین دلیل، هر سازنده گوشی ممکن است مجبور باشد هنگام اعمال تغییرات، سیاست های مجوز مربوط به هر بخش را رعایت کند، که این خود به یک فرآیند استراتژیک تبدیل می شود.

استراتژی تمایز در اینجا می تواند به سادگی تمرکز بر بخش هایی از **Android** باشد که تحت **Apache ۲,۰** قرار دارند، به ویژه در **userspace** (فضای کاربری)، یعنی جایی که برنامه های **Android** اجرا می شوند. **Userspace** شامل برنامه ها و رابط های کاربری است که کاربران مستقیماً با آن ها تعامل دارند، و از آنجا که این بخش تحت مجوز **Apache ۲,۰** است، تولیدکنندگان می توانند تغییرات خود را بدون نیاز به انتشار کد منبع

### Insecure Design (طراحی ناامن)

تهدیدی است که در سال ۲۰۲۱ به فهرست OWASP اضافه شد و بر خطرات ناشی از نقص های طراحی تمرکز دارد. این موضوع نشان دهنده نیاز به استفاده از الگوهای طراحی ایمن تر، مدل سازی تهدید (Threat Modeling) و معماری مرجع در طول فرآیند DevSecOps (توسعه، امنیت و عملیات) است. طراحی ناامن می تواند پایه و اساس سایر آسیب پذیری ها باشد، زیرا اگر سیستم از ابتدا به طور ایمن طراحی نشود، رفع مشکلات در مراحل بعدی دشوارتر خواهد بود. این رویکرد نیازمند توجه ویژه در مراحل اولیه توسعه است.

### Security Misconfiguration (پیکربندی اشتباه امنیتی)

یکی از شایع ترین آسیب پذیری ها است که اغلب به دلیل استفاده از تنظیمات پیش فرض یا نمایش پیام های خطای بیش از حد دقیق که اطلاعات حساس را فاش می کنند، رخ می دهد. این مشکل می تواند به مهاجمان سرنخ هایی درباره ساختار سیستم بدهد و آن ها را به بهره برداری از ضعف ها تشویق کند. سازمان ها باید تنظیمات سیستم های خود را به طور مداوم بررسی کنند و از پیکربندی های امن و به روز استفاده کنند تا از این تهدید جلوگیری شود.

### Vulnerable and Outdated Components (اجزای آسیب پذیر و قدیمی)

نیز تهدیدی جدی به شمار می رود. کتابخانه ها و چارچوب هایی که توسعه دهندگان از آن ها استفاده می کنند، اگر به روز نباشند، ممکن است حاوی نقص های امنیتی شناخته شده باشند. این اجزا، گرچه کار توسعه را آسان تر می کنند، اما در صورت منسوخ شدن می توانند به نقطه ضعفی برای کل سیستم تبدیل شوند. به همین دلیل، مدیریت دقیق نسخه ها و به روز رسانی منظم این اجزا از اهمیت بالایی برخوردار است.

### Identification and Authentication Failures (خرابی در شناسایی و احراز هویت)

به ضعف هایی در سیستم های ورود اشاره دارد. این نقص ها می توانند به مهاجمان اجازه دهند تا با استفاده از حساب های کاربری، حتی حساب های مدیریتی، به سیستم نفوذ کنند. برای مثال، اگر رمز عبور ضعیفی استفاده شود یا مکانیزم احراز هویت به درستی پیاده سازی نشود، کل سیستم در معرض خطر قرار می گیرد. استفاده از احراز هویت چند مرحله ای می تواند راه حلی مؤثر برای این مشکل باشد.

### Software and Data Integrity Failures (نقص نرم افزار و یکپارچگی داده ها)

به زیرساخت ها و کدهایی اشاره دارد که در برابر نقض یکپارچگی، مانند استفاده از گواهی های قدیمی یا امضای دیجیتال ضعیف، آسیب پذیر هستند. این مشکل می تواند به مهاجمان اجازه دهد تا داده ها یا نرم افزار را دستکاری کنند. حفظ یکپارچگی از طریق به روز رسانی مداوم و استفاده از ابزارهای امنیتی مدرن قابل دستیابی است.

## فصل ۶

## متن باز

## نرم افزار برای فناوری های نوظهور

در این بخش، به بررسی استفاده از نرم افزارهای منبع باز (Open Source)، منبع باز) در فناوری های نوظهور مانند هوش مصنوعی (Artificial Intelligence)، هوش مصنوعی)، یادگیری ماشینی (Machine Learning)، یادگیری ماشینی)، و اینترنت اشیا (Internet of Things)، اینترنت اشیا) پرداخته می شود. این فناوری ها در سال های اخیر به دلیل توانایی هایشان در تغییر شکل صنایع و بهبود زندگی روزمره، توجه زیادی به خود جلب کرده اند. ما در اینجا مفاهیم اصلی پشت (AI) هوش مصنوعی)، (ML) یادگیری ماشینی)، (DL Deep Learning)، یادگیری عمیق)، و (IoT) (اینترنت اشیا) را توضیح می دهیم، برخی از پروژه های برجسته و مهم در هر یک از این حوزه ها را معرفی می کنیم، و سپس به چگونگی پذیرش و استفاده از این فناوری ها در صنایع مختلف نگاهی می اندازیم. هدف این است که درک جامعی از این مفاهیم و کاربردهایشان به شما ارائه دهیم تا بتوانید پتانسیل آن ها را در دنیای واقعی بهتر تصور کنید.

ابتدا بیاید با تعریف AI شروع کنیم. هوش مصنوعی به معنای آموزش دادن به ماشین ها برای شبیه سازی رفتارها و توانایی های تفکر انسانی است. این فناوری به سیستم ها اجازه می دهد تا وظایفی را که به طور سنتی توسط انسان ها انجام می شده، به صورت خودکار و با دقت بالا اجرا کنند. برای روشن تر شدن این موضوع، فرض کنید یک مرکز مرتب سازی متمرکز در یک بازار مواد غذایی داریم که در آن محصولاتی مانند شیر، آبمیوه، و آب باید بر اساس برچسب هایشان طبقه بندی شوند. در چنین سیستمی، AI از یک الگوریتم مبتنی بر قوانین (Rule-Based Algorithm)، الگوریتم مبتنی بر قوانین) استفاده می کند تا اشیاء را در دسته های از پیش تعریف شده قرار دهد. این مثال ساده نشان می دهد که چگونه هوش مصنوعی می تواند سرعت و دقت را در انجام کارهای روزمره افزایش دهد. با این حال، این نوع سیستم ها محدودیت هایی دارند، زیرا تنها بر اساس اطلاعاتی که به آن ها داده شده عمل می کنند و انعطاف پذیری محدودی در برابر تغییرات یا خطاها دارند. اما اینجاست که ML وارد میدان می شود و قابلیت های بیشتری به هوش مصنوعی اضافه می کند.

طبق تعریف شرکت (IBM) آی بی ام)، ML شاخه ای از هوش مصنوعی و علوم کامپیوتر است که بر استفاده از داده ها و الگوریتم ها برای تقلید از روش یادگیری انسان ها تمرکز دارد و به تدریج دقت خود را بهبود می بخشد. در واقع، یادگیری ماشینی به ماشین ها این امکان را می دهد که از تجربه یاد بگیرند، بدون اینکه نیاز باشد برای هر وظیفه به طور صریح برنامه ریزی شوند. بیاید به همان مثال مرکز مرتب سازی برگردیم. در سیستمی که فقط از AI ساده استفاده می کند، اگر برچسب یک محصول اشتباه باشد، آن محصول به اشتباه در دسته بندی نادرست قرار می گیرد. اما ML این مشکل را با رویکردی هوشمندانه تر حل می کند. این فناوری با ایجاد یک طرح اولیه (Baseline)، طرح اولیه) بر اساس ویژگی ها و خصوصیات اشیاء کار خود را آغاز می کند. به عنوان مثال، منطق استخراج ویژگی (Feature Extraction)، استخراج ویژگی) به سیستم کمک می کند تا اشیاء را بر اساس رنگ، اندازه ظرف، یا حتی شکل آن ها تشخیص دهد. سپس این سیستم با استفاده از مجموعه داده های شناخته شده



شکل خیالی ۳-۶ که در متن اصلی ذکر شده، تفاوت بین روند آموزش (**Training**)، آموزش و استنتاج را به صورت بصری نشان می‌دهد. در مرحله آموزش، مدل با استفاده از داده‌های مشخص و شناخته‌شده بارها و بارها بهینه می‌شود تا الگوها را یاد بگیرد، اما در استنتاج، مدل باید این دانش را روی داده‌های جدید و ناشناخته اعمال کند. یک مثال عالی از این فرآیند در دنیای واقعی، اپلیکیشن **Uber Eats** (اوبر ایتس) است. این برنامه از قدرت استنتاج برای پیش‌بینی زمان تحویل غذا استفاده می‌کند. وقتی شما سفارشی ثبت می‌کنید، مدل **ML** داده‌هایی مانند فاصله رستوران، ترافیک فعلی، و زمان آماده‌سازی غذا را در نظر می‌گیرد و به سرعت تخمینی از زمان تحویل ارائه می‌دهد. این پیش‌بینی‌ها به لطف استنتاج در زمان واقعی ممکن شده‌اند و تجربه کاربری را بهبود می‌بخشند. چنین کاربردهایی نشان می‌دهند که چگونه **AI/ML** می‌تواند زندگی روزمره ما را کارآمدتر و هوشمندتر کند.

## فاز ۵: گسترش

آخرین مرحله از گردش کار **AI/ML**، گسترش (**Deployment**)، گسترش است که در آن مدل آموزش‌دیده و تأییدشده روی دستگاه یا سیستم مورد نظر پیاده‌سازی می‌شود. این مرحله نهایی است که در آن شما می‌توانید نحوه عملکرد مدل پس از استقرار را در عمل مشاهده کنید. گسترش مدل به این معناست که مدل از محیط آزمایشگاهی یا شبیه‌سازی خارج شده و در یک محیط واقعی مانند یک اپلیکیشن، دستگاه **IoT (Internet of Things)**، یا سیستم ابری (**Cloud System**)، سیستم ابری به کار گرفته می‌شود. اما این پایان کار نیست؛ پس از گسترش، باید مدل را نظارت (**Monitoring**)، نظارت کنید تا مطمئن شوید که همچنان به درستی عمل می‌کند. گاهی اوقات، شرایط تغییر می‌کنند یا داده‌های جدیدی وارد سیستم می‌شوند که مدل قبلی برای آن‌ها آموزش ندیده است. در چنین مواردی، ممکن است لازم باشد مراحل قبلی را با مجموعه داده‌های جدید (**New Datasets**)، مجموعه داده‌های جدید یا ویژگی‌های جدید (**New Features**)، ویژگی‌های جدید) تکرار کنید تا مدل را مجدداً آموزش دهید (**Retraining**)، آموزش مجدد و به‌روز نگه دارید. یکی از ابزارهای محبوب برای این مرحله، **TensorFlow Serving** (تنسورفلو سروینگ) است که بخشی از اکوسیستم **TensorFlow** (تنسورفلو) محسوب می‌شود. این ابزار به توسعه‌دهندگان اجازه می‌دهد تا مدل‌های **ML** را به راحتی در محیط‌های عملیاتی گسترش دهند و آن‌ها را با سیستم‌های دیگر یکپارچه کنند (**Integration**)، یکپارچه‌سازی. برای مثال، اگر یک شرکت بخواهد یک مدل تشخیص تصویر را در یک اپلیکیشن موبایل پیاده‌سازی کند، **TensorFlow Serving** می‌تواند این مدل را روی سرورهای ابری اجرا کرده و پاسخ‌ها را به اپلیکیشن ارسال کند. این ابزار به دلیل انعطاف‌پذیری و پشتیبانی از مقیاس‌پذیری (**Scalability**)، مقیاس‌پذیری بسیار مورد توجه است و به کاربران امکان می‌دهد تا مدل‌ها را به سرعت به‌روزرسانی کنند یا نسخه‌های مختلف را آزمایش کنند (**A/B Testing**)، آزمایش **A/B**.

### گودو: یک موتور بازی منبع باز برجسته

یکی از بهترین نمونه‌های موتورهای منبع باز در صنعت بازی، **Godot** (گودو) است که تحت مجوز **MIT** (ام‌آی‌تی) منتشر شده است. **Godot** یک محیط توسعه یکپارچه (Integrated Development Environment) برای ساخت بازی‌های دوبعدی (۲D) و سه‌بعدی (۳D) ارائه می‌دهد و به دلیل سادگی و قدرت خود، در میان توسعه‌دهندگان محبوبیت زیادی پیدا کرده است. این موتور از چندین زبان برنامه‌نویسی (Programming Languages) پشتیبانی می‌کند، از جمله **C#** سی شارپ، **C++** سی پلاس پلاس و زبان اختصاصی خود به نام **GDScript** (جی‌دی اسکریپت) که شباهت زیادی به **Python** (پایتون) دارد. **GDScript** به‌ویژه برای تازه‌کارها جذاب است، زیرا یادگیری آن آسان است و به توسعه‌دهندگان اجازه می‌دهد به سرعت ایده‌های خود را پیاده‌سازی کنند.

یکی از بزرگ‌ترین نقاط قوت **Godot**، قابلیت انتشار بازی‌ها (Game Deployment) در پلتفرم‌های مختلف است. این موتور امکان ساخت بازی برای دستگاه‌های موبایلی مانند **Android** (اندروید) و **iOS** (آی‌اواس)، پلتفرم‌های دسکتاپ مانند **Linux** لینوکس، **macOS** (مک‌اواس) و **Windows** (ویندوز)، و حتی کنسول‌های بازی (Gaming Consoles) مانند **Nintendo Switch** نینتندو سویچ، **PlayStation 4** (پلی‌استیشن ۴) و **Xbox One** (ایکس‌باکس وان) را فراهم می‌کند. علاوه بر این، بازی‌ها می‌توانند با استفاده از **WebAssembly** (وب‌اسمبلی) و **HTML** (اچ‌تی‌ام‌ال) به وب صادر شوند، که این امکان را به بازی‌سازان می‌دهد تا آثار خود را مستقیماً در مرورگرها (Browsers) اجرا کنند. این انعطاف‌پذیری، **Godot** را به گزینه‌ای ایده‌آل برای توسعه‌دهندگانی تبدیل کرده که می‌خواهند بازی‌هایشان را به مخاطبان گسترده‌تری برسانند.

جامعه **Godot** نیز نقش مهمی در موفقیت این موتور دارد. در پلتفرم **GitHub** (گیت‌هاب)، افزونه‌ها و ابزارهای متعددی برای **Godot** موجود است که توسط توسعه‌دهندگان سراسر جهان ساخته شده‌اند. این افزونه‌ها به کاربران اجازه می‌دهند تا قابلیت‌های موتور را گسترش دهند و آن را با نیازهای خاص خود سازگار کنند. برای مثال، اگر یک توسعه‌دهنده بخواهد یک سیستم هوش مصنوعی پیچیده‌تر یا یک افکت گرافیکی خاص را به بازی خود اضافه کند، می‌تواند از افزونه‌های موجود استفاده کند یا خود یک افزونه جدید بسازد و آن را با جامعه به اشتراک بگذارد. این همکاری، زمان توسعه را به شدت کاهش می‌دهد و به افراد اجازه می‌دهد تا به‌جای تمرکز بر مسائل فنی، روی خلاقیت و داستان‌گویی متمرکز شوند.

### بنیاد لینوکس و Open ۳D Engine

بنیاد **Linux Foundation** (بنیاد لینوکس) نیز با ورود به حوزه بازی‌های سه‌بعدی (۳D) و فناوری شبیه‌سازی (Simulation Technology)، تأثیر بزرگی بر این صنعت گذاشته است. این بنیاد با ایجاد **Open ۳D Foundation** (بنیاد سه‌بعدی باز)، که به اختصار **O3DF** شناخته می‌شود، پروژه‌ای را آغاز کرده که هدف آن توسعه ابزارهای پیشرفته برای بازی‌های سه‌بعدی است. یکی از اولین پروژه‌های این بنیاد،

به دنبال ارتقای مهارت‌های خود هستند. این موضوع نشان می‌دهد که محبوبیت در **GitHub** تنها به کدنویسی محدود نمی‌شود، بلکه اشتراک دانش نیز می‌تواند تأثیر بزرگی داشته باشد.

**جیک وارتون** (Jake Wharton) با ۶۲,۰۰۰ دنبال‌کننده در رتبه پنجم قرار دارد. او به‌خاطر مشارکت‌های گسترده‌اش در پروژه‌های مختلف شناخته شده است، اما اخیراً فعالیت‌هایش بیشتر در مخازن مرتبط با برنامه **Cash App** (اپلیکیشن نقدی) متمرکز بوده است. **جیک وارتون** در توسعه ابزارها و کتابخانه‌هایی که به بهبود تجربه توسعه‌دهندگان در اکوسیستم اندروید (Android Ecosystem) کمک می‌کنند، شهرت دارد. برای مثال، مشارکت او در پروژه‌هایی مانند **Butter Knife** و **Retrofit**، که ابزارهایی برای ساده‌سازی کدنویسی در اندروید هستند، او را به نامی آشنا در این حوزه تبدیل کرده است. حضور او در این فهرست نشان‌دهنده تنوع پروژه‌های تأثیرگذار در **GitHub** است و اینکه چگونه توسعه‌دهندگان می‌توانند در زمینه‌های مختلف، از وب تا موبایل، تأثیر بگذارند.

### تحلیل تأثیرگذاری در دنیای منبع باز

درک این کاربران برتر و فعالیت‌هایشان، به ما کمک می‌کند تا ببینیم چگونه افراد به **اینفلوئنسر** (Influencer) در دنیای **منبع باز** تبدیل می‌شوند. نکته کلیدی این است که برای تأثیرگذار بودن، تنها مشارکت در پروژه‌ها کافی نیست؛ بلکه باید در پروژه‌های حیاتی (Critical Projects) نقش داشته باشید و ترجیحاً بخشی از حلقه بنیان‌گذاران (Founding Circle) آن باشید. همه این کاربران تعهدات قابل توجهی به پروژه‌های خود نشان داده‌اند، اما صرف نوشتن کد به‌تنها نمی‌تواند آن‌ها را به این جایگاه برساند. استفاده از یک فناوری نوظهور (Emerging Technology) یا خلق ابزاری که نیازهای واقعی جامعه را برآورده کند، اهمیت بسیار بیشتری دارد. برای مثال، **لینوس توروالدز** با ابداع **Git** و **لینوکس**، ابزارهایی را خلق کرد که پایه‌های توسعه نرم‌افزار مدرن را تشکیل می‌دهند. به همین ترتیب، **ایوان یو** با **Vue** یک راه‌حل تازه برای توسعه وب ارائه داد که به سرعت مورد استقبال قرار گرفت. این الگو در مورد **دن گائرون** نیز صدق می‌کند، که با توسعه **Redux** و مشارکت در **React**، ابزارهایی را ارائه داد که به استانداردهایی در توسعه برنامه‌های وب تبدیل شدند. **روآن یی‌فنگ** با آموزش‌هایی که شکاف دانش را پر می‌کنند، نشان داد که تأثیرگذاری می‌تواند از طریق آموزش نیز محقق شود. در نهایت، **جیک وارتون** با تمرکز بر ابزارهای کاربردی برای توسعه‌دهندگان اندروید، جایگاه خود را تثبیت کرد. این تنوع نشان می‌دهد که راه‌های مختلفی برای تأثیرگذاری در **GitHub** وجود دارد، اما همه آن‌ها نیازمند تعهد، نوآوری و پاسخگویی به نیازهای جامعه هستند.

### درس‌هایی برای توسعه‌دهندگان و آینده منبع باز

این کاربران نه تنها به‌عنوان الگوهایی برای توسعه‌دهندگان عمل می‌کنند، بلکه درس‌هایی ارزشمند نیز ارائه می‌دهند. اگر می‌خواهید در دنیای **منبع باز** تأثیرگذار باشید، باید پروژه‌ای را انتخاب کنید که پتانسیل تغییر یا بهبود قابل توجهی داشته باشد. فعالیت مداوم، مانند مشارکت‌های سالانه **لینوس توروالدز** یا **ایوان یو**، نشان‌دهنده اهمیت پایداری است. همچنین، خلق ابزارهایی که بهره‌وری (Productivity) یا تجربه توسعه‌دهندگان را بهبود می‌دهند، مانند **Redux** یا **Cash App**، می‌تواند شما را به یک **اینفلوئنسر** تبدیل کند. در نهایت، آموزش و اشتراک دانش، همان‌طور که **روآن یی‌فنگ** انجام داده، راه دیگری برای تأثیرگذاری است.

کمک می کند تا شرکتی را انتخاب کنید که نه تنها به منبع باز متعهد است، بلکه بستری برای رشد حرفه ای شما فراهم می کند.

### نوآوری های آینده و منبع باز

با نگاهی به آینده و رشد فناوری در یکی دو سال پیش رو، می توان دید که مفاهیمی مانند **metaverse** (متاورس) و **Web ۳,۰** (وب ۳,۰) توجه زیادی را به خود جلب کرده اند. این دو حوزه نوظهور نشان دهنده جهت گیری بعدی فناوری هستند و منبع باز می تواند نقش مهمی در توسعه آن ها ایفا کند. **Web ۳,۰** عمدتاً بر تمرکززدایی و کنترل بیشتر کاربران بر اینترنت تمرکز دارد. این مفهوم به دنبال ایجاد بستری است که در آن داده ها و خدمات به صورت غیرمتمرکز مدیریت شوند و کاربران مالکیت بیشتری بر اطلاعات خود داشته باشند. منبع باز با ارائه ابزارها و پروتکل های شفاف می تواند پایه ای محکم برای تحقق این چشم انداز فراهم کند.

از سوی دیگر، **metaverse** بر تجربه کاربران در دنیای دیجیتال تمرکز دارد و به دنبال ایجاد محیط های مجازی است که افراد بتوانند در آن ها تعامل کنند، کار کنند و سرگرمی داشته باشند. این فضا نیازمند ابزارهای پیشرفته ای برای ساخت، مدیریت و اتصال است که منبع باز می تواند به خوبی آن ها را فراهم کند. برای مثال، پروژه های منبع باز می توانند نرم افزارهایی برای شبیه سازی واقعیت مجازی یا توسعه پلتفرم های تعاملی ارائه دهند که به شرکت ها و افراد اجازه می دهد بدون وابستگی به سیستم های اختصاصی، در متاورس مشارکت کنند.

این نوآوری ها فرصت های بی نظیری برای شرکت ها و افراد فراهم می کنند تا در پروژه های منبع باز مرتبط با این حوزه ها مشارکت کنند. شرکت هایی که از هم اکنون در این زمینه ها سرمایه گذاری کنند، می توانند جایگاه خود را به عنوان پیشگامان فناوری تثبیت کنند و استعدادهای برجسته را جذب کنند. از سوی دیگر، افرادی که مهارت های خود را در این حوزه ها توسعه دهند، می توانند در خط مقدم نوآوری قرار گیرند و به شکل دهی آینده دیجیتال کمک کنند.

### وب ۳,۰ و نقش منبع باز

**Web ۳,۰** (وب ۳,۰) به عنوان سومین نسل از اینترنت شناخته می شود که با استفاده از روش های غیرمتمرکز برای اتصال داده ها، تجربه ای سریع تر و شخصی تر را برای کاربران فراهم می کند. این نسل جدید اینترنت با هدف بازگرداندن کنترل به کاربران و کاهش وابستگی به نهادهای متمرکز طراحی شده است. یکی از اصول اساسی **Web ۳,۰**، باز بودن آن است، به این معنا که با استفاده از نرم افزار **open source** (منبع باز) توسط جامعه ای آزاد و در دسترس توسعه می یابد و به صورت شفاف در معرض دید جهانیان اجرا می شود. این شفافیت و دسترسی آزاد به کد، به توسعه دهندگان و کاربران اجازه می دهد تا اطمینان بیشتری به امنیت و کارایی سیستم داشته باشند و در عین حال به نوآوری های جمعی کمک کنند.

یکی از پروژه های برجسته در این حوزه، **Polkadot** (پولکادات) است که توسط **Web ۳ Foundation** (بنیاد وب ۳) راه اندازی شده است. این بنیاد با هدف تسهیل ایجاد یک اینترنت غیرمتمرکز و کاربرپسند فعالیت می کند. پولکادات یک پروتکل منبع باز است که امکان اتصال و همکاری بین بلاک چین های مختلف را فراهم می کند، به گونه ای که داده ها و تراکنش ها به صورت امن و کارآمد بین آن ها منتقل شوند. این پروژه به عنوان یکی از ستون های اصلی **Web ۳,۰** شناخته می شود و نشان دهنده پتانسیل بالای منبع باز در شکل دهی به آینده اینترنت است. در حال حاضر، بیش از هزار



contact me:

*seilany.ir*  
*learninghive.ir*  
*emperor-os.ir*  
*Hubuntu.ir*  
*predator-os.ir*  
*telegram: @seilany*

HosseinSeilani  
Designer, Developer and Linux sysadmin



Founder and Developer of Emperor-OS, Hubuntu and Predator-OS. I bring significant experience as a Linux/Windows sysadmin and graphical web design, UX/UI to the Open Source Community.



Emperor-OS



Predator-OS