

Android

تهیه شده توسط
تیم تولید محتوای سون لرن



اندروید

مسیر یادگیری برنامه نویسی

فهرست محتوا

۴.....	مقدمه
۶.....	چرا اندروید را انتخاب کنیم؟ (ابتدا هدف را تعیین کنید)
۸.....	چگونه اندروید را یاد بگیریم؟
۹.....	زبان برنامه نویسی جاوا و ویژگی‌های آن
۱۱.....	زبان برنامه نویسی کاتلین و ویژگی‌های آن
۱۳.....	مقایسه کاتلین و جاوا
۱۳.....	کنترل استثناها در کاتلین
۱۴.....	سادگی و اختصار
۱۴.....	مدیریت NullPointerException
۱۵.....	تعامل بین کاتلین و جاوا
۱۵.....	برای توسعه اپلیکیشن‌های اندروید جاوا بهتر است یا کاتلین؟
۱۷.....	بالاخره جاوا یا کاتلین؟
۱۸.....	منابع یادگیری برنامه نویسی اندروید
۱۹.....	نقشه راه یادگیری برنامه نویسی اندروید
۲۰.....	۱- آموزش برنامه نویسی
۲۰.....	۲- آموزش اندروید استودیو
۲۲.....	۳- کامپوننت‌های اندروید
۲۴.....	۴- طراحی رابط کاربری ساده
۲۵.....	۵- طراحی رابط کاربری پیچیده
۲۶.....	۶- ذخیره سازی
۲۶.....	۷- Build
۲۷.....	۸- Threading

- ۹- اشکال زدایی..... ۲۷
- ۱۰- کمبود یا نشت حافظه ۲۷
- ۱۱- کتابخانه‌های شخص ثالث ۲۸
- ۱۲- Android Jetpack ۲۹
- ۱۳- معماری اندروید..... ۳۰
- ۱۴- فایربیس..... ۳۰
- ۱۵- تست واحد..... ۳۱
- ۱۶- امنیت..... ۳۱
- ۱۷- انتشار برنامه..... ۳۱
- پیشنهاد سون‌لرن برای شروع..... ۳۲

مقدمه

اندروید یک سیستم عامل منبع باز مبتنی بر هسته لینوکس است که در دستگاه‌هایی همچون تلفن‌های هوشمند، تبلت‌ها، ساعت‌های هوشمند، Android TV و... استفاده می‌شود. هر یک از این دستگاه‌ها رابط کاربری مختص خود را دارند و تنها در سیستم عامل، مشترک هستند.

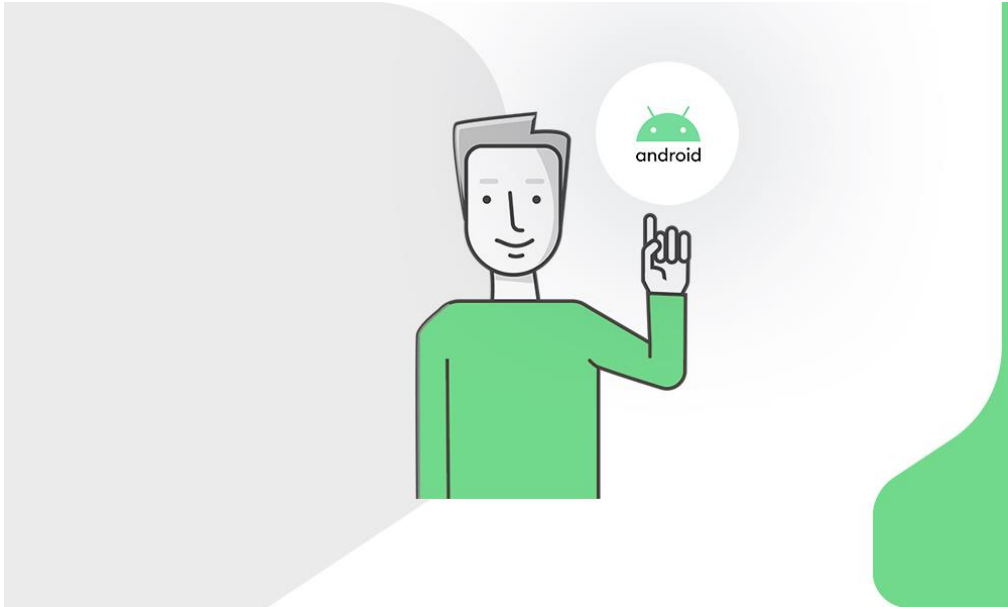
در سال‌های اخیر، سیستم عامل اندروید یکی از پرفروش‌ترین سیستم عامل‌ها برای گوشی‌های هوشمند بوده است. امروزه سهم استفاده مردم جهان از سیستم عامل اندروید تقریباً ۷۵٪ است در حالی که تنها ۱۵٪ از آن‌ها سیستم عامل iOS را ترجیح می‌دهند.

سیستم عامل اندروید توسط Android Inc، که گوگل در سال ۲۰۰۵ آن را خریداری کرده بود، توسعه داده شده است. در حال حاضر، در فروشگاه Google Play بیش از ۳.۳ میلیون برنامه، اعم از بازی‌ها، پخش کننده موسیقی، دوربین و... برای

گوشی‌های اندرویدی موجود است که نشان می‌دهد بازار هدف اندروید تا چه حد گسترده و متنوع است.

محبوبیت بالای سیستم عامل اندروید، کاربردهای گسترده و بازار کار خوب آن، موجب شده است که سوالات بسیاری در ذهن علاقه‌مندان به یادگیری برنامه نویسی شکل بگیرد. چرا اندروید را انتخاب کنم؟ برنامه نویسی اندروید را از کجا شروع کنم؟ چه مباحثی را باید یاد بگیرم؟ برنامه نویسی اندروید را با مطالعه‌ی کتاب یاد بگیرم یا آموزش‌های آنلاین و انجام پروژه؟ ما در این کتاب قصد داریم به تمامی این سوالات پاسخ دهیم. اگر در تصمیم خود برای یادگیری برنامه نویسی جدی و ثابت قدم هستید، تبریک می‌گوییم! آینده روشنی در انتظار شماست. کتاب پیش رو نیز شما را در مسیر صحیح یادگیری اندروید هدایت می‌کند.

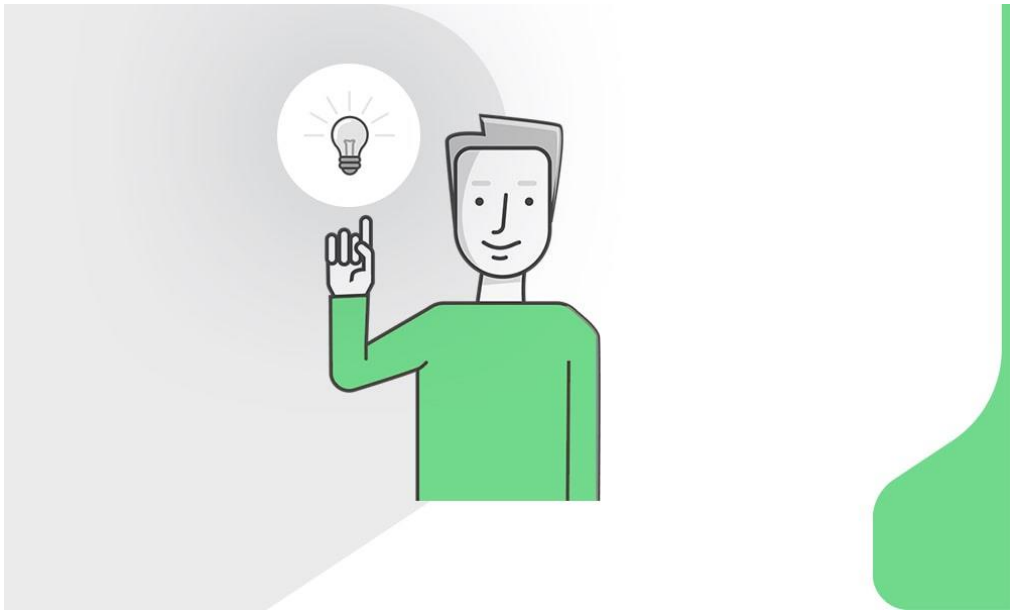
چرا اندروید را انتخاب کنم؟ (ابتدا هدف را تعیین کنید)



پیش از ورود به نقشه راه کامل یادگیری برنامه نویسی اندروید، باید یک هدف روشن در ذهن خود داشته باشید. ابتدا تعیین کنید که هدف شما از یادگیری برنامه نویسی اندروید چیست؟ آیا هدفتان انجام پروژه‌های دانشگاهی است؟ یا اینکه برنامه نویسی اندروید را به عنوان شغل آینده خود می‌بینید؟ یا می‌خواهید برنامه‌های خود را برای شروع کارتان توسعه دهید؟ هدف شما می‌تواند مسیر متفاوتی را برایتان تعیین کند. به عنوان مثال، اگر می‌خواهید اندروید را برای انجام پروژه‌های

دانشگاهی خود یاد بگیرید، کافیست تنها مباحث ابتدایی اندروید را بیاموزید. به همین ترتیب، اگر به برنامه نویسی اندروید به چشم حرفه آینده خود نگاه می‌کنید، باید مباحث حرفه‌ای یا پیشرفته را نیز یاد بگیرید. این تصمیم با شماست که چرا می‌خواهید اندروید را یاد بگیرید.

چگونه اندروید را یاد بگیریم؟



در اندروید، برنامه نویسی با استفاده از دو زبان انجام می‌شود: JAVA یا Kotlin و XML (Extension Markup Language). طراحی فرانت اند یک برنامه اندرویدی (طراحی، نحوه ارائه، طرح بندی، نقشه ساخت (blueprint) و...) با XML و توسعه بک اند با JAVA یا KOTLIN (ساخت دکمه‌ها، متغیرها، ذخیره سازی و...) انجام می‌شود. بزرگترین سردرگمی برای یک مبتدی اندروید این است که کدام زبان را بین جاوا و کاتلین انتخاب کند؟ بنابراین بهتر است ابتدا به بررسی این موضوع بپردازیم.

زبان برنامه نویسی جاوا و ویژگی‌های آن



جاوا زبان بسیار قدرتمندی است که برنامه نویسان زیادی در سراسر جهان از آن استفاده می‌کنند. می‌توان گفت جاوا به شما این توانایی را می‌دهد که در هر زمینه‌ای برنامه نویسی کنید و با بسیاری از محیط‌های نرم افزاری و سخت افزاری همخوانی دارد. در برنامه‌های تحت ویندوز، لینوکس و همچنین توسعه اپلیکیشن‌های اندروید می‌توانید از جاوا استفاده کنید.

همچنین چندین سال است که این زبان در بسیاری از موسسه‌ها و دانشگاه‌های جهان تدریس می‌شود. بالا بودن

تعداد برنامه نویسان جاوا باعث شده این زبان انجمن‌های (Community) بسیار گسترده‌ای در سطح جهان داشته باشد.

برخی از ویژگی‌های جاوا عبارت‌اند از:

- امکان مازوله سازی کدها و استفاده مجدد در بخش‌های دیگر از پروژه
- کارایی بالا و پشتیبانی از امنیت نرم افزارها
- مستقل از پلتفرم و آسان برای یادگیری

زبان برنامه نویسی کاتلین و ویژگی‌های آن



کاتلین یکی از زبان‌های جدید برنامه نویسی است که سال ۲۰۱۱ توسط JetBrains معرفی شد. این زبان که به یکی از رقبای اصلی جاوا در توسعه اندروید تبدیل شده در طی این سال‌ها محبوبیت خوبی به دست آورده و توانسته تا حدودی جاوا را به دومین گزینه برنامه نویسان تبدیل کند. هدف توسعه دهندگان این زبان، سریع‌تر کردن روند توسعه، برای برنامه نویسانی است که بخواهند از زبانی غیر از جاوا استفاده کنند.

برخی ویژگی‌های کاتلین عبارت‌اند از:

- امکان استفاده از کتابخانه‌ها و فریم‌ورک‌های جاوا
- امکان تبدیل کدهای جاوا به کاتلین به طور اتوماتیک
- کنترل راحت‌تر Null Exceptions

مقایسه کاتلین و جاوا



حال بیایید نگاهی به تفاوت‌های این دو زبان محبوب بیندازیم و ببینیم برنامه نویسان طبق چه معیارهایی باید جاوا یا کاتلین را برای توسعه کدهای خود استفاده کنند.

کنترل استثناها در کاتلین

استثناها (Exceptions) یکی از موارد مشکل‌ساز در جاوا هستند که در بسیاری از موارد برنامه نویسان را درگیر می‌کنند. اما کاتلین به خوبی این مشکل را مدیریت کرده است. این

زبان کنترل تمامی اکسپشن‌ها را به دست گرفته و دیگر برنامه نویسان نیازی به کنترل این موارد در کدهای خود ندارند.

سادگی و اختصار

اینکه بتوان با تعداد خطوط کد کمتر، بخش زیادی از پروژه را پیاده سازی کرد مسئله‌ای است که در پروژه‌های بزرگ اهمیت زیادی دارد. برخلاف جاوا، دستورات کاتلین به طور خلاصه و مفید به شما کمک می‌کنند تا روند توسعه نرم افزار را سریع‌تر کنید. کاتلین دستورات بسیار راحتی دارد و می‌تواند شروع خوبی برای یادگیری برنامه نویسی باشد. کسانی که آشنایی چندانی با برنامه نویسی ندارند نیز، می‌توانند به یادگیری کاتلین بپردازند.

مدیریت NullPointerException

استثناهای مقادیر Null، یکی از خطاهای رایج در جاوا هستند که حتی ممکن است بسیاری از برنامه نویسان با انواع مختلف

آن آشنا نباشند. کاتلین این مشکل را حل کرده و برنامه نویسان را با چنین خطایی مواجه نمی‌کند.

تعامل بین کاتلین و جاوا

جاوا یک زبان قدیمی و بسیار قوی است که نمی‌توان به راحتی آن را نادیده گرفت. کاتلین با تکیه بر جاوا توسعه پیدا کرده و این قابلیت را دارد که کدهای جاوا را نیز پوشش دهد. در کاتلین می‌توان از کتابخانه‌های جاوا استفاده کرد. همچنین می‌توان کدهای آن را، درون کدهای جاوا قرار داد.

برای توسعه اپلیکیشن‌های اندروید جاوا بهتر است یا کاتلین؟

ممکن است شما در شروع راه برنامه نویسی باشید. جاوا یکی از گزینه‌های خوب در توسعه اندروید است و شما را قادر می‌سازد پروژه‌های متنوع دیگری را نیز انجام دهید اما بیشترین استفاده از کاتلین، در برنامه نویسی اندروید است.

پس با توجه به اهدافی که دارید، یادگیری هر دو زبان می‌تواند مزایای زیادی برای شما داشته باشد.

حال اگر از قبل با زبان جاوا آشنایی دارید شاید بهتر باشد کار توسعه اپلیکیشن‌های اندروید را با این زبان شروع کنید. جاوا منابع زیادی برای یادگیری دارد و شما می‌توانید در بسیاری از موارد پاسخ سوال‌های خود را به راحتی در انجمن‌های فعال آن پیدا کنید. علاوه بر این کلاس‌های آموزشی و افراد باتجربه زیادی نیز در حوزه برنامه نویسی جاوا فعالیت می‌کنند.

کاتلین هنوز هم منابع نسبتاً کمی دارد و مانند جاوا فراگیر نشده است. پس اگر تسلط خوبی بر جاوا دارید، یاد گرفتن کاتلین برای شما کار سختی نیست. وقت آن است که به سراغ کاتلین رفته و از این زبان محبوب در پروژه‌های خود استفاده کنید. یاد گرفتن کاتلین برای کسانی که بر جاوا تسلط دارند مزیت بزرگی محسوب می‌شود و شرکت‌ها را برای استخدام آن‌ها وسوسه می‌کند.

یکی از اشتباهات درباره زبان کاتلین این است که این زبان در پروژه‌های بزرگ کارایی ندارد اما این تصور اشتباه است. کاتلین با کاهش حجم کد و مختصر کردن دستورات، می‌تواند برای پروژه‌های بزرگ نیز انتخاب خوبی باشد.

بالاخره جاوا یا کاتلین؟

هر دو زبان ویژگی‌های منحصر به فرد خود را دارند و به طور قطعی نمی‌توان هیچکدام را حذف کرد. جاوا که همواره پر قدرت است و کاتلین نیز تا کنون توانسته طرفداران خود را راضی نگه دارد. توصیه ما به مبتدیان این است که ابتدا به سراغ یادگیری جاوا و سپس کاتلین بروید. چرا کاتلین را یاد بگیریم؟ زیرا گوگل، کاتلین را به عنوان زبانی درجه یک اعلام کرده است. پس اگر کاتلین زبانی قدرتمند است، پس چرا جاوا را یاد بگیریم؟ چون برای کسب بیشترین بهره از کاتلین، باید دانش جاوا داشته باشیم. بنابراین ابتدا با جاوا شروع کنید، سپس کاتلین را بیاموزید و مهارت‌های خود را ارتقا دهید.

منابع یادگیری برنامه نویسی اندروید

منابع و فیلم‌های زیادی به صورت آنلاین در دسترس هستند و ممکن است برای افرادی که در ابتدای مسیر یادگیری اندروید هستند، گیج کننده باشد که از کجا باید همه مفاهیم را به صورت یکجا یاد بگیرند. در ابتدای این راه، شما با حجم زیادی از اصطلاحات و مفاهیم رو به رو می‌شوید اما نباید بترسید و یادگیری را متوقف کنید. صبر داشته باشید، کاوش کنید و به آن پایبند باشید. اگر به زبان انگلیسی تسلط نسبتاً خوبی دارید، توصیه می‌کنیم از منابع زیر در وب سایت GeeksforGeeks استفاده کنید:

[Java Programming Language](#)

[Kotlin Programming Language](#)

[XML](#)

[Android](#)

نقشه راه یادگیری برنامه نویسی اندروید

ابتدا با شناخت اندروید شروع کنید. مطالب وبلاگ‌های مرتبط با اندروید را بخوانید. به عنوان مثال وبلاگ‌هایی در مورد مقدمه‌ای بر توسعه اندروید، تاریخچه اندروید، نسخه‌های مختلف اندروید و همچنین موضوعاتی مانند چرا کاتلین جایگزین جاوا برای توسعه برنامه‌های اندروید است و ... ، بخوانید و یک آرایش کامل ذهنی برای شروع سفر خود در اندروید ایجاد کنید. برای یادگیری اندروید و ساخت چند پروژه عالی در اندروید به خودتان انگیزه دهید. این کار را به طور منظم انجام دهید و همچنین یادگیری مفاهیم جدید را شروع کنید. قبل از شروع سفر، بهتر است در چند وبینار یا کنفرانس در اندروید نیز شرکت کرده و با افراد و حوزه‌ی تخصصی اندروید بیشتر آشنا شوید. هدف خود را روشن کنید و به سمت آن پیش بروید.

در ادامه مراحل یادگیری برنامه نویسی اندروید را توضیح داده‌ایم:

۱- آموزش برنامه نویسی

ممکن است برخی افراد، این مرحله را به عنوان پیش نیاز در نظر بگیرند. قبل از شروع یادگیری Android، این زبان‌های برنامه نویسی را یاد بگیرید:

- زبان جاوا
- زبان کاتلین
- دانش نسبی از زبان نشانه گذاری XML

۲- آموزش اندروید استودیو

بهتر است در قدم‌های آغازین، ابزارهای موردنیاز خود را بشناسید. Android Studio محیط رسمی توسعه یکپارچه برای سیستم عامل Android است که بر روی نرم افزار IntelliJ IDEA JetBrains ساخته شده و به طور خاص برای توسعه Android طراحی شده است.

ساختار فایل:

فایل AndroidManifest.xml

فایل Java

فایل Drawable

فایل Layout

فایل mipmap

فایل colors.xml

فایل strings.xml

فایل themes.xml

فایل build.gradle(Module: app)

بررسی Android Studio:

یک پروژه جدید ایجاد کنید.

پروژه را دوباره باز کنید، ببندید، ذخیره کنید.

یک فعالیت جدید، چند کلاس و فایل‌های مرجع مختلف ایجاد کنید.

برنامه را روی AVD Emulator یا در یک دستگاه واقعی اجرا کنید.

۳- کامپوننت‌های اندروید

یک اپلیکیشن اندرویدی از تعدادی بلوک‌های ساختاری تشکیل شده است. این کامپوننت‌های جفت آزاد با فایل مانیفست اپلیکیشن محدود شده‌اند که شامل توضیح هر کامپوننت و نحوه تعامل با آن است.

اکتیویتی:

چرخه عمر اکتیویتی

کنترل تغییرات حالت اکتیویتی

درک Tasks و Back Stack

فرآیندها و چرخه عمر اپلیکیشن

سرویس‌ها:

انواع سرویس‌های اندروید

چرخه عمر سرویس‌های اندروید

ارائه دهنده محتوا:

URI محتوا

عملیات در ارائه دهنده محتوا

کار ارائه دهنده محتوا

ایجاد یک ارائه دهنده محتوا

گیرنده پخش:

استثنائات پخش ضمنی

۴- طراحی رابط کاربری ساده

بعد از این که کامپوننت‌های مختلف اندروید را درک کردید، در مورد طراحی رابط کاربری ساده که در زیر آورده شده است، جستجو کنید.

FrameLayout

LinearLayout

RelativeLayout

ConstraintLayout

عناصر مشاهده:

TextView

EditText

Buttons

ImageView

:Intent

Implicit

Explicit

Intent Filter

۵- طراحی رابط کاربری پیچیده

ListView

RecyclerView

Fragments

Dialogs

Toast

Snackbar

Bottom Sheets

Navigation Drawer

Tabs

Material Design

انیمیشن‌های سرگرم کننده

۶- ذخیره سازی

در اندروید، چهار نوع سیستم ذخیره سازی وجود دارد:

Shared Preferences

File System

Database

RoomDB

Build - ۷

Gradle

Debug/ Release Configuration

Threading - ۸

Threads

Looper

۹- اشکال زدایی

یکی از مهم‌ترین مهارت‌های یک توسعه دهنده، مهارت اشکال زدایی است. بنابراین توسعه دهنده باید این موارد را بیاموزد:

Exceptions

Error Handling

Logging

Memory Profiling

۱۰- کمبود یا نشت حافظه

دلایل نشت حافظه

تشخیص و رفع نشت حافظه

کانتکس

۱۱- کتابخانه‌های شخص ثالث

کتابخانه‌های بارگذاری تصویر:

Glide

Picasso

Fresco

COIL

تزریق وابستگی:

Dragger

Hilt

Koin

شبکه سازی:

Retrofit

:Multithreading

Coroutines

Rxjava

۱۲- Android Jetpack

Jetpack راهکاری جدید برای ساخت اپلیکیشن‌های اندرویدی، به روشی ساده‌تر، بی دردسرتر، تست پذیرتر و توسعه پذیرتر است.

Compose

AppCompat library

Architecture components

Animation and transitions

Android Ktx

Navigation

Paging

Slices

WorkManager

۱۳- معماری اندروید

سه معماری معروف در دنیای اندروید عبارت‌اند از:

MVVM (Model-View-ViewModel)

MVI (Model-View-Intent)

MVP (Model View Presenter)

۱۴- فایربیس

فایربیس (Firebase) یک پلتفرم قدرتمند است که برای

ساخت اپلیکیشن‌های وب و موبایل مقیاس پذیر و امن

استفاده می‌شود. موارد ضروری فایربیس عبارت‌اند از:

FCM (Firebase Cloud Messaging)

Analytics

Remote Config

App Indexing

۱۵- تست واحد

هدف از تست واحد این است که مشخص کنیم که آیا هر کدام از واحدها به درستی کار می‌کنند یا خیر.

Local Unit Testing

Instrumentation Testing

۱۶- امنیت

Encrypt / Decrypt

Proguard

۱۷- انتشار برنامه

Signed APK

Play Store

پیشنهاد سون لرن برای شروع

اگر به دنبال یک روش جامع و تضمینی برای یادگیری اندروید و دست یافتن به شغل دلخواهتان هستید، دوره متخصص اندروید آکادمی سون لرن را پیشنهاد می‌کنیم.

در این دوره، که طبق به‌روزترین سرفصل‌ها تدریس می‌شود، دانشجویان از مزایای زیر بهره‌مند می‌شوند:

- آموزش مبتنی بر انجام تمرین، پروژه و آزمون
- ارزیابی فنی در انتهای دوره
- پشتیبانی تا ۳ ماه بعد از دوره
- مشاوره ورود به بازار کار
- معرفی دانشجویان برتر به شرکت‌های معتبر و متقاضی

برای شرکت در دوره از لینک زیر اقدام نمایید.

بیشتر بخوانید: [دوره متخصص اندروید](#)

در پایان از اینکه برای خواندن این کتاب وقت گذاشته‌اید، از شما سپاس‌گزاریم و در وبسایت و شبکه‌های اجتماعی سون‌لرن آماده دریافت نظرات و پیشنهادات شما هستیم.

از یادگیری تا ورود به بازار کار

www.7Learn.com