
برنامه نویسی کیوی به زبان ساده (برنامه نویسی اندروید با پایتون)

تألیف:

دکتر رمضان عباس نژادورزی
دکتر محمد احمدزاده



فن آوری نوین

شماره کتابشناسی ملی	:	ایران ۸۴۱۳۱۸۵
شابک	:	۵-۴۱-۷۳۹۳-۶۲۲-۹۷۸-۹۵۰۰۰۰ ریال
سرشناسه	:	عباس‌نژاد ورزی، رمضان، ۱۳۴۸ -
عنوان و نام پدیدآور	:	برنامه‌نویسی کیوی به زبان ساده [منابع الکترونیکی: کتاب]: برنامه‌نویسی اندروید با پایتون/تألیف رمضان عباس‌نژاد ورزی، محمد احمدزاده.
مشخصات نشر	:	بابل: فناوری نوین، ۱۴۰۰.
مشخصات ظاهری	:	۱ منبع بر خط (۲۸۴ص.).: مصور (بخشی رنگی).
وضعیت فهرست نویسی	:	فبیا
نوع منبع الکترونیکی	:	فایل Pdf.
یادداشت	:	دسترسی از طریق وب.
عنوان دیگر	:	برنامه‌نویسی اندروید با پایتون.
شناسه افزوده	:	احمدزاده، محمد، ۱۳۶۸ -
موضوع	:	ترم افزار کیوی
موضوع	:	Kivy (Computer software*)
موضوع	:	اندروید (منبع الکترونیکی) -- برنامه‌نویسی
موضوع	:	Android (Electronic resource) -- Programming
موضوع	:	پایتون (زبان برنامه‌نویسی کامپیوتر)
موضوع	:	Python (Computer program languag)
رده بندی کنگره	:	۷۶/۷۶QA
رده بندی دیویی	:	۰۰۵/۲۷۶۸
دسترسی و محل الکترونیکی	:	آدرس الکترونیکی منبع

@fanavarienovinpub

تلفن: ۰۱۱-۳۲۲۵۶۶۸۷

بابل، کد پستی ۴۷۱۶۷-۷۳۴۴۸

فن آوری نوین

برنامه‌نویسی کیوی به زبان ساده

تألیف: رمضان عباس‌نژاد ورزی، محمد احمد زاده

نوبت چاپ: چاپ اول

سال چاپ: تابستان ۱۴۰۰

شمارگان: ۲۰۰

قیمت: ۹۵۰۰۰ تومان

نام چاپخانه و صحافی: دفتر فنی سورنا

شابک: ۵-۴۱-۷۳۹۳-۶۲۲-۹۷۸

نشانی ناشر: بابل، چهارراه نواب، کاظم‌بیگی، جنب مسجد منصور کاظم‌بیگی، طبقه اول

طراح جلد: کانون آگهی و تبلیغات آبان (احمد فرجی)

فروشگاه و پخش کتاب چاپی: تهران، تلفن ۶۶۴۰۰۲۲۰ - ۶۶۴۰۰۱۴۴

تهران، خ اردیبهشت، نبش وحید نظری، پلاک ۱۴۲ تلفکس: ۶۶۴۰۰۱۴۴-۶۶۴۰۰۲۲۰

فهرست مطالب

فصل اول: آشنایی با کتابخانه کیوی	۷
۱-۱. مقدمه‌ای بر برنامه‌نویسی اندروید با پایتون	۷
۱-۲. پایتون یک زبان برنامه‌نویسی بسیار ساده و ظریف	۷
۱-۳. پایتون چیست؟	۷
۱-۴. چرا بهتر است برنامه‌نویسی اندروید با پایتون انجام شود؟	۸
۱-۵. نقش پایتون در توسعه موبایل	۸
۱-۶. برنامه‌های کاربردی با استفاده از برنامه‌نویسی اندروید با پایتون	۹
۱-۷. برنامه‌نویسی اندروید با استفاده از کتابخانه Kivy	۹
۱-۸. Kivy چیست؟	۱۰
۱-۹. مشخصات کلیدی kivy	۱۱
۱-۱۰. معایب برنامه‌نویسی اندروید با پایتون	۱۲
۱-۱۱. معماری کیوی	۱۲
۱-۱۲. نصب بسته Kivy	۱۷
۱-۱۳. نوشتن اولین برنامه با Kivy	۱۹
۱-۱۴. معرفی ویجت‌ها	۲۱
۱-۱۴-۱. ویجت label	۲۲
۱-۱۴-۲. ویجت Button	۲۶
۱-۱۴-۳. ویجت TextInput	۳۸
فصل دوم: مقدمه‌ای بر زبان برنامه‌نویسی کیوی	۴۳
۲-۱. بررسی زبان کیوی	۴۳
۲-۲. نحو یک فایل kv	۴۳
۲-۳. مقدار عبارات، عبارات on_property، شناسه‌ها و کلمات کلیدی رزرو شده	۴۶
۲-۴. اضافه کردن مازول‌ها و کلاس‌های پایتون از طریق kv	۴۸
۲-۵. مقداردهی به یک متغیر یا کلید از طریق kv	۴۸
۲-۶. معرفی کردن ویجت فرزند	۴۸
۲-۷. انقیاد رویداد	۴۹
۲-۸. عبارات معتبر	۵۰
۲-۹. رابطه بین مقادیر و خواص	۵۰
۲-۱۰. دستورات گرافیکی	۵۲
۲-۱۱. کلاس‌های دینامیک	۵۳
۲-۱۲. الگوها	۵۴
۲-۱۳. تعریف مجدد سبک ویجت	۵۷
۲-۱۴. تعریف مجدد سبک ویژگی ویجت	۵۸
۲-۱۵. ترتیب kwargs و برنامه قانون KV	۵۸
۲-۱۶. دستورات Lang	۵۹
۲-۱۷. دستور import <package>	۵۹

۶۰.....	۲-۱۸. دستور set <key> <expr>.....
۶۰.....	۲-۱۹. دستور include <file>.....
۶۱.....	۲-۲۰. بررسی چند فایل kv.....
۷۷.....	فصل سوم: چیدمان عناصر رابط کاربری
۷۷.....	۳-۱. رابط کاربری.....
۷۷.....	۳-۲. چیدمان FloatLayout.....
۸۲.....	۳-۳. چیدمان RelativeLayout.....
۸۵.....	۳-۴. چیدمان GridLayout.....
۹۲.....	۳-۵. چیدمان BoxLayout.....
۹۹.....	۳-۶. چیدمان StackLayout.....
۱۰۳.....	۳-۷. چیدمان AnchorLayout.....
۱۰۷.....	۳-۸. چیدمان ScatterLayout.....
۱۱۰.....	۳-۹. ترکیب چیدمان‌ها.....
۱۲۳.....	فصل چهارم: گرافیک در کیوی (بوم)
۱۲۳.....	۴-۱. مقدمه.....
۱۲۳.....	۴-۲. انواع دستورالعمل‌های کار بر روی Canvas.....
۱۲۳.....	۴-۲-۱. زیر کلاس‌های VertexInstruction.....
۱۴۳.....	۴-۲-۲. زیر کلاس‌های ContextInstructions.....
۱۵۳.....	فصل پنجم: ویجت‌های طراحی واسط کاربری
۱۵۳.....	۵-۱. مقدمه.....
۱۵۳.....	۵-۲. ویجت Checkbox.....
۱۶۲.....	۵-۳. ویجت ToggleButton.....
۱۶۸.....	۵-۴. ویجت Switch.....
۱۷۱.....	۵-۵. ویجت Image.....
۱۷۳.....	۵-۶. ویجت Slider.....
۱۸۰.....	۵-۷. ویجت Spinner.....
۱۸۳.....	۵-۸. ویجت Popup.....
۱۸۸.....	۵-۹. ویجت ProgressBar.....
۱۹۴.....	۵-۱۰. ویجت Drop-Down List.....
۱۹۷.....	۵-۱۱. ویجت TabbedPanel.....
۲۰۲.....	۵-۱۲. ویجت TreeView.....
۲۱۰.....	۵-۱۳. ویجت ColorPicker.....
۲۱۴.....	۵-۱۴. ویجت FileChooser.....
۲۲۵.....	فصل ششم: پروژه‌های برنامه‌نویسی
۲۸۴.....	منابع:

مقدمه

با محبوبیت گوشی‌های هوشمند و تبلت‌ها، برنامه‌نویسان طالب برنامه‌نویسی اندروید با پایتون شده‌اند. به لطف کتابخانه جدید کیوی، برنامه‌نویسان پایتون می‌توانند برنامه‌های اندرویدی زیادی را توسعه دهند.

برای این که بتوانیم از پایتون برای برنامه‌نویسی اندروید استفاده کنیم، باید از کتابخانه برنامه‌نویسی با عنوان Kivy بهره بگیریم. این کتابخانه یک پلتفرم چند سکویی می‌باشد که برای برنامه‌نویسی اندروید با پایتون استفاده می‌شود. قبل از این که شروع به استفاده از این کتابخانه‌ی برنامه‌نویسی کنیم، باید حتماً اقدام به نصب کتابخانه‌ی Kivy نماییم.

کیوی در **برنامه‌نویسی اندروید** یک کتابخانه‌ای است که به شما اجازه می‌دهد با استفاده از پایتون برای کامپیوترهای شخصی و موبایل‌ها برنامه‌های کاربردی طراحی کنید. این کتابخانه کدهای شما را دریافت می‌کند و با استفاده از منطق شما برنامه کاربردی موردنظر را ایجاد می‌کند. سپس فایل apk فوق را تولید می‌کند.

یکی از ویژگی‌های بارز kivy در این است که یک بار در آن کد نویسی می‌کنید و سپس می‌توانید در چندین پلتفرم متفاوت از آن استفاده کنید و یا آن را توسعه دهید.

با kivy در برنامه‌نویسی اندروید با پایتون شما برای هر پلتفرم نیاز به توسعه‌دهنده یا برنامه‌نویس ندارید. پس در برنامه‌نویسی اندروید با پایتون نیازی نیست که یک برنامه را چندین بار در پلتفرم‌های متفاوت طراحی کنیم. یعنی، برنامه یک بار برای تمامی پلتفرم‌ها طراحی می‌کنیم. با این اوصاف برخی از ویژگی‌های بارز کتابخانه کیوی عبارت‌اند از:

- ❖ Kivy، یک کتابخانه رایگان و متن‌باز هست.
 - ❖ Kivy، یک محیط گرافیکی برای ساخت اپلیکیشن موبایل فراهم می‌کند.
 - ❖ با Kivy، می‌توانید برنامه‌هایی بنویسید که دارای فرمان‌های چند لمسی هستند.
 - ❖ با Kivy، می‌توانید برنامه‌هایی برای موبایل بنویسید که هم روی اندروید و هم روی iOS اجرا شوند.
 - ❖ Kivy را می‌شود روی ویندوز، لینوکس، مک و Raspberry Pi هم اجرا کرد.
- کتابخانه Kivy شامل تمام امکانات برای ساخت نرم‌افزارهای مختلف است که برخی از این امکانات عبارت‌اند از:

- ❖ پشتیبانی کامل از ورودی‌های ماوس، صفحه‌کلید و ورودی‌های لمسی مخصوص سیستم‌عامل.
 - ❖ از کتابخانه گرافیک اوپن جی ال (OpenGL) پشتیبانی می‌کند.
 - ❖ دارای ویجت‌های گرافیکی بسیار گسترده است که از تکنولوژی چند لمسی پشتیبانی می‌کنند.
 - ❖ دارای زبان برنامه‌نویسی KV است که برای راحتی در توسعه واسط کاربری طراحی شده است.
- از تمامی اساتید و دانشجویان عزیز تقاضا داریم، هرگونه اشکال، ابهام در متن کتاب، پیشنهاد و انتقادات را به آدرس پست الکترونیک fanavarienovin@gmail.com ارسال نمایند.
- در پایان امیدوارم این اثر مورد توجه جامعه انفورماتیک کشور، اساتید و دانشجویان عزیز قرار گیرد.

fanavarienovin@gmail.com

مؤلفین - تابستان ۱۴۰۰

مرجع کامل

برنامه نویسی پایتون

تألیف :

« دکتر جواد وحیدی

عضو هیات علمی دانشگاه علم و صنعت ایران

« دکتر رمضان عباس نژاد ورزی

آشنایی با کتابخانه کیوی

۱-۱. مقدمه‌ای بر برنامه‌نویسی اندروید با پایتون

برنامه‌نویسی یک مهارت فوق‌العاده‌ای برای یادگیری است و اندروید امکاناتی بیش‌تری را در این زمینه در اختیار برنامه‌نویس قرار می‌دهد و محدودیت‌های زیادی را در این زمینه حذف کرده است که یک گزینه عالی برای شروع **برنامه‌نویسی اندروید** می‌باشد.

مسئله اینجا است که کد نویسی کردن تحت اندروید زیاد هم کار آسانی نیست. قبل از این که بخواهید یک برنامه ساده را بنویسید نیاز دارید نرم‌افزار android studio را همراه با jdk و بقیه امکانات آن دانلود کرده و نصب کنید. سپس باید بدانید که چگونه بایستی یک فایل APK ایجاد کنید و مجوزهای مربوط به آن را اعمال کرده و روی گوشی خودتان نصب کنید. حتی قبل از این که بتوانید برای خودتان خروجی بگیرید و در صفحه‌نمایش آن را ببینید باید با یک سری از بخش‌ها در این نرم‌افزار آشنا بشوید.

۱-۲. پایتون یک زبان برنامه‌نویسی بسیار ساده و ظریف

بنا به دلایل فوق ممکن است برای شما بسیار جذاب به نظر بیاید. پایتون یک زبان برنامه‌نویسی بسیار ساده و ظریف است که با ساختاری ساده طراحی شده است.

مهم‌تر از این کامپایل کردن کدهای شما و تست کردن آن‌ها روی دستگاه اندروید می‌تواند بسیار سریع انجام شود. پایتون یک زبان برنامه‌نویسی یکی از سریع‌ترین روش‌ها می‌باشد تا بتوانید برنامه خودتان را تولید و به اجرا برسانید. علاوه بر این، این‌ها زمانی که شما در **برنامه‌نویسی اندروید با پایتون** با ویژگی‌های بیش‌تری آشنا می‌شوید و آن‌ها را به کار می‌برید می‌توانید از ترفندهای شگفت‌انگیز زیادی در جهت امکانات بخشی به تلفن همراه خود استفاده کنید. در **برنامه‌نویسی اندروید با پایتون** کافی است فقط با چند حرکت کوتاه یک برنامه APK کامل ایجاد کنید.

۱-۳. پایتون چیست؟

پایتون یک زبان برنامه‌نویسی نسبتاً جدید است که توسط گیدو ون راسوم در سال ۱۹۹۱ طراحی شده است. فلسفه رایج طراحی این زبان قابل فهم بودن آن است. به عبارتی دیگر، یعنی این که حتی یک غیر کد نویس هم بتواند کدهای آن را پیگیری کرده و بفهمد.

پایتون فضای سفید زیادی را مورد استفاده قرار می‌دهد و باعث استفاده کارآمدی از دستورات می‌شود. این یعنی با تعداد خطوط کم‌تری از دستورات می‌تواند کارهای زیادی را انجام دهد.

۸ فصل اول

این سادگی و ظرافت باعث می‌شود پایتون انتخاب خوبی برای برنامه‌نویسان جدید باشد. برای شروع، مفسران قابل‌دسترسی در سیستم‌عامل‌های چندوظیفه‌ای وجود دارند. یعنی، می‌توانید کدهای خودتان را در ویندوز، مک، لینوکس و اندروید اجرا کنید.

پایتون یکی از زبان‌های برنامه‌نویسی اصلی است که شامل طیف گسترده‌ای از ابزارهای هیجان‌انگیز می‌باشد که این زبان را یک زبان ایده آل برای یادگیری کودکان می‌کند.

۴-۱. چرا بهتر است برنامه‌نویسی اندروید با پایتون انجام شود؟

دلایلی زیادی وجود دارد که برنامه‌نویسی اندروید با پایتون کاراتر به نظر می‌رسد. برخی از این دلایل عبارت‌اند از:

- ❖ پایتون توانایی اجرا در هریک از سیستم‌عامل‌های مهم از قبیل لینوکس، یونیکس، ویندوز، مک و غیره را دارد که می‌تواند گزینه مناسبی برای **برنامه‌نویسی اندروید** باشد.
- ❖ زبان پایتون، زبانی با مفهوم و هدف‌دار ارائه می‌کند تا به شما اجازه دهد برنامه‌هایی با مقیاس کوچک و بزرگ به‌صورت رایگان و بدون محدودیت توسعه دهید که در این‌جا نتیجه می‌گیریم برنامه‌نویسی اندروید با پایتون بسیار ساده انجام می‌گیرد.
- ❖ انواع ابزارهای ارائه‌شده توسط پایتون برای هر دو گروه توسعه‌دهندگان و مدیران سیستم وجود دارد. یعنی در **برنامه‌نویسی اندروید با پایتون** تنوع ابزار زیاد است.
- ❖ با توجه به توانایی پایتون برای انعطاف‌پذیر بودن و پویا بودن، کاربرانی مانند گوگل، یاهو و آی بی ام از کار کردن با آن لذت می‌برند. پس می‌توان گفت که برنامه‌نویسی اندروید با پایتون جالب‌تر خواهد بود.
- ❖ نرخ انتقال و سرعت توسعه در پایتون بسیار بالا می‌باشد که اجازه می‌دهد برنامه‌های مشابه در تمامی پلتفرم‌ها عمل کنند.
- ❖ پایتون برای کار بر روی یک وظیفه خاصی شامل کتابخانه‌های غنی و بسیاری از بسته‌های دیگر است. پس محدودیتی در توابع نداریم.

۵-۱. نقش پایتون در توسعه موبایل

توسعه برنامه‌های موبایل به دلیل گسترش دامنه آن به بخش عمده‌ای از کسب‌وکار تبدیل شده است. فریم‌ورک پلتفرم پایتون در بسیاری از سیستم‌عامل‌ها از قبیل اندروید، ویندوز، لینوکس و مک اجرا می‌شود.

آشنایی با کتابخانه کیوی ۹

پایتون یک ابزار عالی برای نوشتن اسکریپت‌های ساده و برنامه‌های پیچیده **چند فغی** می‌باشد. نکته قابل توجهی که در **برنامه‌نویسی اندروید با پایتون** وجود دارد آن است که این فرصت را به دست آورده است تا به صورت رایگان در دسترس باشد و در سطرهای نامحدود در آن کد نویسی انجام می‌شود. برنامه‌نویسان دستگاه‌های اندرویدی و آیفون برای ارائه یک برنامه تلفن همراه عالی به مشتری خود، از تعدادی تکنیک توسعه چند سکویی (یعنی قابلیت اجرا در چندین سیستم عامل) استفاده می‌کنند.

۶-۱. برنامه‌های کاربردی با استفاده از برنامه‌نویسی اندروید با پایتون

برنامه‌های زیادی در حوزه موبایل با پایتون پیاده‌سازی شده‌اند که برخی از مهم‌ترین آن‌ها عبارت‌اند از:

- ❖ **برنامه Aarlogic C05/3** : یک برنامه ردیابی PCB از GSM/GPS استفاده می‌کند که با پایتون توسعه یافته است و با سرویس دهنده‌های نقشه گوگل مورد پشتیبانی قرار گرفته است.
- ❖ **برنامه Pyrout** : یک برنامه نقشه برداری و مسیریابی GPS مناسب برای موبایل می‌باشد.
- ❖ **برنامه FoodPlus** : یک برنامه موبایل سفارش غذا می‌باشد که فرآیند سفارش غذا و ردیابی را ساده می‌کند و مخصوص علاقه‌مندان به غذا با استفاده از برنامه‌نویسی اندروید با پایتون طراحی شده است.
- ❖ **برنامه AppBackup** : این برنامه دستگاه‌های IOS را جیلبریک می‌کند که باعث می‌شود امکانات پشتیبانی و بازیابی تنظیمات و داده‌های برنامه فروشگاه موبایل فعال شوند.

۷-۱. برنامه‌نویسی اندروید با استفاده از کتابخانه Kivy

با محبوبیت گوشی‌های هوشمند و تبلت‌ها، برنامه‌نویسان طالب برنامه‌نویسی اندروید با پایتون شده‌اند. به لطف کتابخانه جدید، برنامه‌نویسان پایتون می‌توانند در برنامه‌نویسی اندروید با پایتون با استفاده از **کتابخانه kivy** برنامه‌های زیادی را توسعه دهند. در زمینه برنامه‌نویسی اندروید با پایتون به دانشی پایه نیاز دارید. در این بخش ویژگی‌های کلیدی Kivy را معرفی می‌شوند.

برای این که بتوانیم از پایتون برای برنامه‌نویسی اندروید استفاده کنیم، باید از کتابخانه برنامه‌نویسی با عنوان Kivy بهره بگیریم. این کتابخانه یک پلتفرم چند سکویی می‌باشد که برای برنامه‌نویسی اندروید با پایتون استفاده می‌شود. قبل از این که شروع به استفاده از این کتابخانه‌ی برنامه‌نویسی کنیم، باید حتماً اقدام به نصب کتابخانه‌ی Kivy نماییم. ممکن است نصب کتابخانه‌ی Kivy تا حدودی زمان‌بر و وقت گیر باشد، اما برای توسعه نرم‌افزار اندرویدی مورد نظر کاملاً ضروری می‌باشد. در ادامه نصب کتابخانه kivy را در ویندوز مشاهده می‌کنید.

۸-۱. Kivy چیست؟

کیوی در برنامه‌نویسی اندروید یک کتابخانه‌ای است که به شما اجازه می‌دهد با استفاده از پایتون برای کامپیوترهای شخصی و موبایل‌ها برنامه‌های کاربردی طراحی کنید. این کتابخانه کدهای شما را دریافت می‌کند و با استفاده از منطق شما برنامه کاربردی موردنظر را ایجاد می‌کند. سپس فایل apk فوق را تولید می‌کند.

❖ یک بار برنامه‌نویسی انجام می‌شود و چندین بار آن را گسترش می‌دهیم.

یکی از ویژگی‌های بارز Kivy در این است که یک بار در آن کد نویسی می‌کنید و سپس می‌توانید در چندین پلتفرم متفاوت از آن استفاده کنید و یا آن را توسعه دهید. هرکسی که تلفن هوشمندی دارد ممکن است با این مشکل روبه‌رو شود که یک برنامه شگفت‌انگیز در دستگاهی دیگر ببیند و نتواند آن را برای گوشی مخصوص خودش پیدا کند. حتی بیش‌تر مواقع شایع می‌شود که یک برنامه مفید موبایلی تولید می‌شود که نمونه دسک‌تاپ آن موجود نمی‌باشد. در این مواقع، هرکسی ممکن است دارای گوشی هوشمند نباشد یا برنامه‌ای را خواستار باشد که با تمامی دستگاه‌ها و در همه زمان‌ها سازگار باشد.

❖ یک بار برای تمامی پلتفرم‌ها طراحی می‌کنیم

با Kivy در برنامه‌نویسی اندروید با پایتون شما برای هر پلتفرم نیاز به توسعه‌دهنده یا برنامه‌نویس ندارید. فقط یک برنامه‌نویس کافی است تا در تمامی این پلتفرم‌ها برنامه موردنظر را تولید کند و به‌صورت پکیج‌های جداگانه برای هر برنامه فروشگاهی دستگاه‌ها استخراج کند. پس در برنامه‌نویسی اندروید با پایتون نیازی نیست که یک برنامه را چندین بار در پلتفرم‌های متفاوت طراحی کنیم. یعنی، یک بار برای تمامی پلتفرم‌ها طراحی می‌کنیم. با این اوصاف برخی از ویژگی‌های بارز کتابخانه کیوی عبارت‌اند از:

❖ Kivy یک کتابخانه رایگان و متن‌باز هست.

❖ Kivy یک محیط گرافیکی برای ساخت اپلیکیشن موبایل فراهم می‌کند.

❖ با Kivy می‌توانید برنامه‌هایی بنویسید که دارای فرمان‌های چند لمسی هستند.

❖ با Kivy می‌توانید برنامه‌هایی برای موبایل بنویسید که هم روی اندروید و هم روی iOS اجرا شوند.

❖ Kivy را می‌شود روی ویندوز، لینوکس، مک و Raspberry Pi هم اجرا کرد.

کتابخانه Kivy شامل تمام امکانات برای ساخت نرم‌افزارهای مختلف است که برخی از این امکانات عبارت‌اند

از:

❖ پشتیبانی کامل از ورودی‌های ماوس، صفحه‌کلید و ورودی‌های لمسی مخصوص سیستم‌عامل.

❖ کتابخانه گرافیک فقط شامل اوپن جی‌ال (OpenGL).

آشنایی با کتابخانه کیوی ۱۱

- ❖ ویجت‌های گرافیکی بسیار گسترده که از تکنولوژی چند لمسی پشتیبانی می‌کنند.
- ❖ زبان برنامه‌نویسی اختصاصی Kv که برای راحتی در توسعه محیط کاربری طراحی شده است.
- ❖ کیوی ارتقا یافته PyMT project است و برای پروژه‌های جدید پیشنهاد می‌شود.

۹-۱. مشخصات کلیدی kivy

Kivy در برنامه‌نویسی اندروید با پایتون برای برنامه‌های موبایل و دسکتاپ فراتر از یک چارچوب می‌رود. آن می‌تواند با سخت‌افزارهای تمامی دستگاه‌ها ارتباط برقرار کند و حتی رندرهای پیشرفته‌ای را انجام دهد. در برنامه‌نویسی اندروید با پایتون، کیوی به موتورهای رندر دسترسی دارد و حتی می‌تواند خیلی فراتر از یک برنامه وب ساده عمل کند.

❖ **دوربین:** برای گوشی‌های تلفن‌های هوشمند امکان وفق دهنده‌گی در اجرا دارد. این یکی از نخستین الزاماتی است که ما را وادار می‌کند به جای یک برنامه وب پاسخ‌گو، برنامه‌ای نیتیو ایجاد کنیم. اگرچه در html5 برای دوربین‌ها در مرورگرها در بهترین حالت نقطه‌ضعفی وجود دارد.

❖ **ویدئو:** Kivy شامل ماژول‌هایی برای خواندن و پخش فیلم‌ها می‌باشد. درست است که این کار را یک برنامه وب هم می‌تواند انجام دهد، اما تجربه نشان داده است که گاهی اوقات ممکن است ویدئو کیفیتش ضعیف شود. درحالی‌که که گوشی‌های قدیمی ممکن است در این حالت ویدئو را بدون اشغال تمام صفحه پخش کنند یا حافظه‌ای که قرار است برای پخش این ویدئو استفاده شود بالاتر از سطح انتظار رود.

❖ **چند لمسی و چند حرکتی:** Kivy دارای یک ماژول قوی برای ورودی است که اجازه می‌دهد تا چند لمسی و چند حرکتی در برنامه خود را پیاده‌سازی کنید.

❖ **جلوه‌های بصری و open GL:** از آنجایی که Kivy در برنامه‌نویسی اندروید با پایتون به OpenGL دسترسی دارد، می‌تواند آرایه وسیعی از جلوه‌های بصری را به صورت کامل با صرف رندر کمی انجام دهد. کیوی همچنین از PyGame استفاده می‌کند، بنابراین ایجاد بازی‌ها بسیار ساده‌تر می‌شود (اگرچه باید کد PyGame موجود را برای Kivy بازنویسی کنید. PyGame دارای ماژول‌هایی برای طراحی اشکال، رندرینگ، برخورد با رنگ‌ها و پخش موسیقی می‌باشد).

❖ **درخواست‌های شبکه غیر هم‌زمان:** یکی از ویژگی‌های مهم در هر برنامه‌های بومی حداقل برای افرادی مثل من که در داخل و خارج از ناحیه شبکه سفر می‌کنند، توانایی تکمیل درخواست‌ها هنگامی که یک شبکه در دسترس است، می‌باشد. یعنی، دوست دارم که زمانی که وارد تونل شدم هم روی گوشی خود بتوانم کار

انجام دهم و سپس هنگامی که به یک ایستگاهی رسیدم بتوانم تمام کارهای خودم را به همگام سازی کنم و بر روی بانک اطلاعاتی ابری ثبت شوند.

۱۰-۱. معایب برنامه نویسی اندروید با پایتون

در اینجا به مهم ترین معایب برنامه نویسی اپلیکیشن با پایتون اشاره می کنیم که عبارت اند از:

❖ **مشکل در استفاده از کدهای زبان های دیگر،** توسعه اپلیکیشن ها عموماً با زبان های نیتیو انجام می شود. زبان نیتیو برای اندروید جاوا و کاتلین است و زبان نیتیو برای ios سویت است. درست است که می توان از سایر زبان های دیگر برای توسعه اپ استفاده کرد اما گاهی نیاز از که قسمتی از کد به صورت نیتیو نوشته شود. پایتون چنین توانایی ندارد.

❖ **عملکرد ضعیف در محاسبات روی موبایل،** پایتون برای توسعه اپلیکیشن ها و نرم افزارها روی دستگاه های دسکتاپ و سرویس دهنده توسعه داده شده است. در نتیجه، روی دسکتاپ و سرویس دهنده ها عملکرد خیلی خوبی دارد اما روی موبایل عملکرد ضعیفی در محاسبات دارد. به همین علت اپلیکیشن های کمی با پایتون توسعه داده شده است.

❖ **کندتر بودن در اجرا،** پایتون به صورت مفسر اجرا می شود نه به صورت کامپایلر. این امر باعث کاهش سرعت اپلیکیشن های توسعه داده شده با پایتون می شود. چون کامپایل شدن و سپس اجرای برنامه باعث افزایش سرعت اجرای اپلیکیشن می شود.

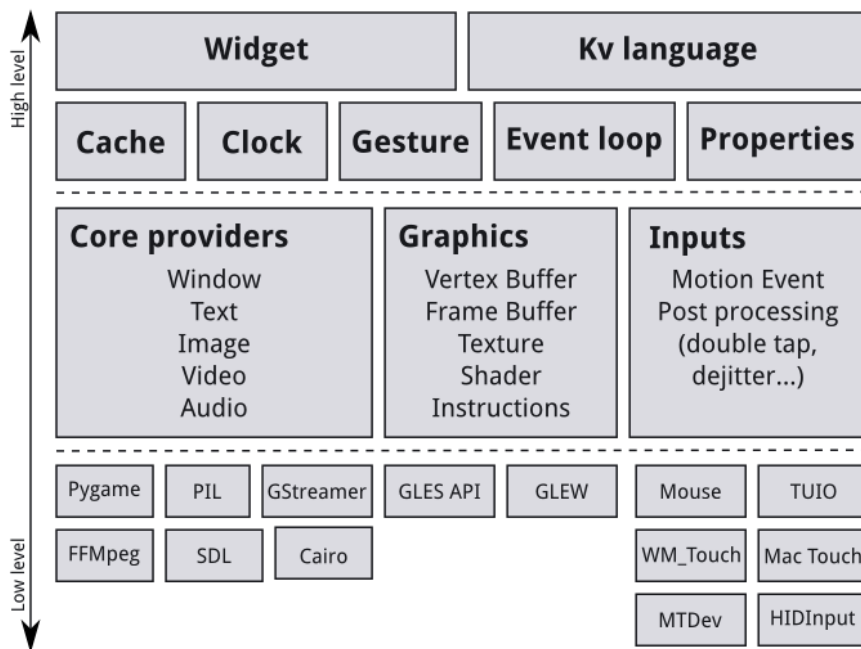
❖ **خطاهای اجرا،** بسیاری از خطاهای پایتون در زمان اجرا ظاهر می شود. به همین علت توسعه اپلیکیشن ها با پایتون نیازمند زمان و تست های بیش تری است

۱۱-۱. معماری کیوی

قبل از یادگیری برنامه نویسی کیوی ابتدا باید با مهندسی فریم ورک آن آشنا شویم. در این بخش شرح خواهیم داد که فریم ورک کیوی از لحاظ معماری و مهندسی نرم افزار چگونه طراحی شده است. اگر فقط به کد نگاه کنید، به احتمال زیاد یک ایده تقریبی خواهید گرفت، اما چون این روش مطمئناً برای اکثر کاربران به ویژه کاربران تازه کار دلهره آور است، برای درک بهتر کدهایی که در محیط کیوی پیاده سازی می کنید، بهتر است با معماری کیوی آشنا باشید. به همین دلیل، در این بخش ایده های اساسی پیاده سازی را با جزئیات بیش تری توضیح می دهیم. معماری Kivy از چندین بلوک تشکیل شده است که در شکل ۱-۱ خلاصه ای از آن آورده شده است. این شکل از بخش های زیر تشکیل شده است:



Kivy Architecture



شکل ۱-۱ معماری Kivy.

ارائه‌دهندگان اصلی و ارائه‌دهندگان ورودی (Core Providers and Input Providers)

ارائه‌دهندگان اصلی در مهندسی فریم‌ورک کیوی

یک ایده که برای درک بهتر معماری داخلی کیوی مهم است، ایده مدولاراسیون و انتزاع است. سعی می‌کنیم کارهای اساسی مانند باز کردن پنجره، نمایش تصاویر و متن، پخش صدا، گرفتن تصاویر از دوربین، تصحیح املائی و غیره را انتزاع کنیم. این‌ها را **وظایف اصلی** می‌نامیم. این امر باعث می‌شود که API به‌راحتی قابل استفاده باشد و همچنین به‌راحتی قابل توسعه است. از همه بهتر، این کار باعث می‌شود کیوی به ما اجازه دهد تا از تأمین‌کننده‌ها ناظر هر سناریویی که اپ شما در حال اجرای آن است، استفاده شود. برای مثال، در OS X، windows، linux، android و API‌های متفاوت و بومی برای اجرای وظایف اصلی یا اولیه وجود دارد. یک کد که از API خاصی برای صحبت کردن با سیستم عامل از یک طرف و kivy از طرف دیگر استفاده می‌کند که اصطلاحاً به آن ارائه‌دهنده اصلی یا هسته (core provider) گفته می‌شود.

۱۴ فصل اول

مزایای استفاده از ارائه‌دهنده‌های اصلی تخصصی در مهندسی فریم‌ورک کیوی برای هر پلت‌فرم به شرح زیر می‌باشد:

۱. عملکرد سیستم عامل بالا می‌رود.

۲. سیستم کارآمدتر می‌شود.

این اقدام به کاربر یک حق انتخاب می‌دهد. به علاوه با استفاده از کتابخانه‌هایی که در هر پلت‌فرم وجود دارد به صورت مؤثر اندازه توزیع kivy کاهش می‌یابد و بسته‌بندی به آسانی انجام می‌شود. با این اقدام همچنین انتقال kivy به پلت‌فرم‌های دیگر راحت‌تر انجام می‌شود. پلت‌فرم‌های اندروید به صورت گسترده‌ای از این روش بهره‌مند شده‌اند.

ارائه‌دهنده‌های ورودی در مهندسی فریم‌ورک کیوی

از همین مفهوم برای مدیریت ورودی پیروی می‌کنیم. ارائه‌دهنده ورودی قطعه‌ای از کد است که پشتیبانی از یک دستگاه ورودی خاص مانند trackpads اپل، TUIO یا شبیه‌ساز ماوس را اضافه می‌کند. اگر نیاز به پشتیبانی از دستگاه ورودی جدید دارید، به راحتی می‌توانید یک کلاس جدیدی اضافه کنید. این کلاس داده‌های ورودی شما را می‌خواند و آن‌ها را به داده‌های پایه‌ای kivy تبدیل می‌کند. حال که فهمیدید این ارائه‌دهنده‌ها چه هستند، می‌خواهیم با چند تا از این ارائه‌دهنده‌ها در زمینه‌های مختلف و این که چرا در مهندسی فریم‌ورک کیوی به کار برده شده‌اند، آشنا شویم.

ارائه‌دهنده گرافیکی در مهندسی فریم‌ورک کیوی

فریم‌ورک کیوی سرویس‌های گرافیکی خود را از طرف API رندر گرافیکی OpenGL ES می‌گیرد که در ساده‌ترین بخش کیوی دستورات ترسیم سخت‌افزاری را با استفاده از OpenGL ES صادر می‌کند. نوشتن کد OpenGL اما به‌ویژه برای تازه‌واردان می‌تواند کمی سخت و گیج‌کننده باشد. به همین دلیل، کیوی API گرافیکی ارائه می‌دهد که به آسانی با استفاده از استعاره‌هایی مانند Canvas و Rectangle می‌توانید شکل رسم کنید.

همه ویجت‌های kivy خود از این API گرافیکی استفاده می‌کنند که به دلایل عملکرد در سطح C اجرا می‌شود. یکی دیگر از مزیت‌های API گرافیکی توانایی آن در بهینه‌سازی خودکار دستورات طراحی است که کد شما آن‌ها را صادر می‌کند. این امر به‌ویژه در صورتی که متخصص تنظیم OpenGL نباشید، بسیار مفید است. این موضوع باعث می‌شود در بسیاری از موارد کد ترسیم شما کارآمدتر شود. البته اگر ترجیح می‌دهید هنوز می‌توانید از دستورات خام OpenGL استفاده کنید و همان‌طور که ذکر شد برای افراد نوپا بسیار سخت و پیچیده است. نسخه موردنظر ما در همه دستگاه‌ها (OpenGL ES ۲.۰ و GLES۲) است. بنابراین، اگر می‌خواهید سازگار با کراس پلت‌فرم باشید، به شما توصیه می‌کنیم فقط از توابع GLES۲ استفاده کنید.

ارائه‌دهنده هسته (core) در مهندسی فریم‌ورک کیوی

این بسته کد شامل چندین ارائه‌دهنده مختلف می‌باشد که این ارائه‌دهنده‌ها از آن قبیل افزونه‌ها می‌باشند که در بخش نصب فریم‌ورک کیوی توضیح خواهیم داد.

ساعت (Clock)

یکی از مجموعه کدهایی که سرویس‌های زمان را می‌دهد و جزو افزونه‌هایی است که امکان دارد به هیچ وجه روی ویندوز نصب نشود یا حتی در صورتی که نصب شود اصلاً کار نکند. ولی کاربرد اصلی این ماژول برای برنامه‌ریزی و زمان‌بندی هر چیز به خصوص اجرای کدها است که این تایمرها به صورت دوره‌ای پشتیبانی می‌شوند.

حافظه پنهان (Cache)

زمانی که چیزی را همیشه استفاده می‌کنید نیازی نیست آن را مدام بنویسید. بلکه می‌توانید از کلاس‌های ما استفاده کنید.

تشخیص حرکت (Gesture Detection)

این ماژول برای تشخیص حرکت است که کیوی ارسال می‌کند. این ماژول در مهندسی فریم‌ورک کیوی آن-قدر پیشرفته عمل می‌کند که در صورتی که به خوبی آموزش ببیند حتی می‌تواند با رسم و پردازش اشکالی چون مستطیل و دایره استفاده کنید. می‌توانید آن را آموزش دهید تا انواع سگته‌های مغزی را تشخیص دهد.

زبان کیوی (kivy language)

بسته کد زبان کیوی یکی از پکیج‌هایی است که اکثر مواقع به طور پیش فرض با کیوی بر روی سیستم نصب می‌شود. از بسته کدهای زبان کیوی برای توصیف واسطه‌های آسان و کارآمد استفاده می‌شود (فصل دوم به این موضوع می‌پردازیم). این‌ها خصوصیات عادی نیستند که شاید از پایتون بدانید. آن‌ها کلاس‌های خاص خود ما هستند که کد ویجت شما را با توضیحات رابط کاربر پیوند می‌دهند.

خواص

این بسته کد تنظیمات عادی از پایتون نیست که فکر کنید بدانید. بلکه آن‌ها کلاس‌های خاص خود پایتون هستند که کدها و ویجت‌های شما را با توضیحات رابط کاربر پیوند می‌دهند.

UIX (ویجت‌ها و چیدمان‌ها)

ماژول UIX شامل ویجت‌ها و چیدمان‌هایی است که معمولاً از آن‌ها استفاده می‌کنید تا به سرعت بتوانید یک رابط کاربری ایجاد کنید.

ویجت‌ها (widgets): عناصر رابط کاربری هستند که شما به برنامه خود اضافه می‌کنید تا نوعی کارایی ارائه شود. آن‌ها ممکن است دیده شوند یا نشوند. نمونه‌های ویجت‌ها می‌توانند یک مرورگر فایل، دکمه‌ها، اسلایدرها، لیست‌ها و غیره باشند. ویجت‌ها MotionEvents را دریافت می‌کنند.

۱۶ فصل اول

چیدمان‌ها (layouts): برای چینش ویجت‌ها یا ابزارک‌ها (widgets) از چیدمان یا لایوت (layouts) استفاده می‌کنید.

مطمئنأ می‌توانید موقعیت‌های ویجت خود را محاسبه کنید، اما اغلب استفاده از یکی از لایوت‌های آماده ما راحت‌تر است. نمونه‌های آن می‌تواند لایوت گرید (grid layouts) یا لایوت جعبه (box layouts) باشد. همچنین می‌توانید چیدمان‌های لایه را تهیه کنید.

ماژول‌ها (modules)

اگر تاکنون از یک مرورگر وب مدرن استفاده کرده‌اید و آن را با برخی از افزودنی‌ها سفارشی کرده‌اید، از قبل ایده اصلی کلاس‌های ماژول کیوی را می‌دانید. ماژول‌ها را می‌توان برای تزریق عملکرد به برنامه‌های Kivy استفاده کرد، حتی اگر نویسنده اصلی آن را شامل نشود. به‌عنوان مثال، ماژولی در معماری فریم‌ورک کیوی خواهد بود که همیشه تعداد فریم بر ثانیه (FPS (frame per second)) برنامه فعلی و برخی نمودارها را که FPS را به‌مرورزمان نشان می‌دهد، نشان می‌دهد. همچنین می‌توانید ماژول‌های خود را بنویسید.

کلیدهای ورودی (لمسی) ((Input Events (touch)) در مهندسی فریم‌ورک کیوی

فریم‌ورک کیوی kivy انواع مختلفی از منابع ورودی دارد که شامل ورودی لمسی (touch)، TUIO، ماوس و غیره را در برمی‌گیرد. البته آن چیزی که همه‌ی این ورودی‌ها را با هم مشترک می‌کند، این است که می‌توانید یک موقعیت دوبعدی (D2) در صفحه ایجاد کنید و با هر ورودی فردی منطبق کنید. دستگاه‌های ورودی دیگری مانند شتاب‌سنج وجود دارد که شما به‌راحتی نمی‌توانید موقعیت دوبعدی به‌عنوان مثال شیب دستگاه خود را پیدا کنید. این نوع ورودی به‌طور جداگانه اداره می‌شود.

همه این نوع ورودی‌ها توسط نمونه‌هایی از کلاس Touch() نمایش داده می‌شوند. توجه داشته باشید که این موضوع فقط به لمس انگشت اشاره نمی‌کند، بلکه انواع دیگر ورودی نیز وجود مثل ماوس دارد. فقط به خاطر سادگی آن را لمس نام می‌بریم. برای چیزی که رابط کاربری یا صفحه‌نمایش شما آن را لمس کنید. یا موضوع می‌تواند در یکی از سه حالت باشد. وقتی یک لمس وارد یکی از این حالت‌ها می‌شود، به برنامه شما اطلاع داده می‌شود که این رویداد رخ داده است. سه حالت لمسی که در آن قرار دارد عبارت‌اند از:

❖ **پایین (Down) در kivy:** تعداد لمس فقط یک‌بار است و در همان لحظه‌ای که برای اولین بار ظاهر می‌شود، پایین می‌رود که به‌طور مثال انگار شما کلیدی را به سمت پایین فشار دادید.

❖ **حرکت (Move) در kivy:** لمس می‌تواند در این حالت برای مدت‌زمان نامحدودی باشد. حرکت درزمانی که موقعیت دوبعدی یک لمس تغییر کند، اتفاق می‌افتد.

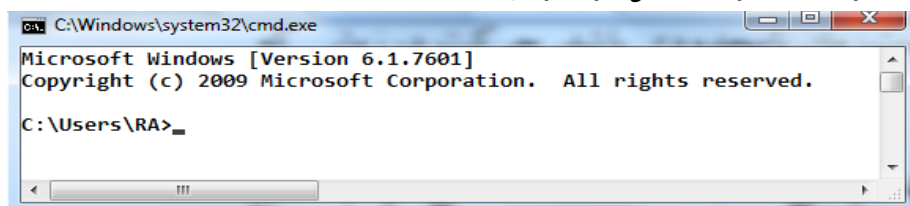
آشنایی با کتابخانه کیوی ۱۷

❖ **بالا (Up) در kivy**: لمس حداکثر یک بار، یا هرگز اتفاق نمی‌افتد. زمانی که به گفته ساده شما می‌خواهید تأثیر کلید را بردارید که ورودی به برنامه می‌دهد یا عملیاتی انجام می‌دهد. دست از لمس برمی‌دارید یا دوباره کلید را فشار می‌دهید که کلید اصطلاحاً بالا بیاید.

۱۲-۱. نصب بسته Kivy

اولین قدم برای برنامه‌نویسی با kivy نصب آن است. در این بخش نصب کتابخانه kivy را در سیستم عامل ویندوز بیان می‌کنیم. برای این منظور، مراحل زیر را انجام دهید:

۱. به خط فرمان (cmd) بروید تا شکل زیر ظاهر شود:



۲. دستور `pip install kivy` را اجرا کنید تا پس از مدتی کتابخانه kivy نصب گردد (مانند شکل زیر):

```
C:\Windows\system32\cmd.exe
C:\Users\RA>pip install kivy
Collecting kivy
  Downloading Kivy-2.0.0-cp38-cp38-win32.whl (3.8 MB)
    |████████████████████████████████████████| 3.8 MB 242 kB/s
Collecting kivy-deps.angle<=0.3.0
  Downloading kivy_deps.angle-0.3.0-cp38-cp38-win32.whl (4.2 MB)
    |████████████████████████████████████████| 4.2 MB 22 kB/s
Collecting kivy-deps.sdl2<=0.3.1
  Downloading kivy_deps.sdl2-0.3.1-cp38-cp38-win32.whl (2.3 MB)
    |████████████████████████████████████████| 2.3 MB 210 kB/s
Collecting pywin32
  Downloading pywin32-223-py3-none-any.whl (1.7 kB)
Collecting docutils
  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
    |████████████████████████████████████████| 548 kB 252 kB/s
Collecting Kivy-Garden>=0.1.4
  Downloading kivy-garden-0.1.4.tar.gz (6.8 kB)
Collecting Pygments
  Downloading Pygments-2.8.0-py3-none-any.whl (983 kB)
    |████████████████████████████████████████| 983 kB 243 kB/s
Collecting kivy-deps.glew<=0.3.0
  Downloading kivy_deps.glew-0.3.0-cp38-cp38-win32.whl (126 kB)
    |████████████████████████████████████████| 126 kB 252 kB/s
Collecting requests
  Downloading requests-2.25.1-py2.py3-none-any.whl (61 kB)
    |████████████████████████████████████████| 61 kB 230 kB/s
Collecting pywin32>=223
  Downloading pywin32-300-cp38-cp38-win32.whl (8.5 MB)
    |████████████████████████████████████████| 8.5 MB 345 kB/s
Collecting chardet<5,>=3.0.2
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
    |████████████████████████████████████████| 178 kB 328 kB/s
Collecting certifi>=2017.4.17
  Downloading certifi-2020.12.5-py2.py3-none-any.whl (147 kB)
    |████████████████████████████████████████| 147 kB 345 kB/s
Collecting idna<3,>=2.5
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
    |████████████████████████████████████████| 58 kB 250 kB/s
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.3-py2.py3-none-any.whl (137 kB)
    |████████████████████████████████████████| 137 kB 364 kB/s
Using legacy 'setup.py install' for Kivy-Garden, since package 'wheel' is not installed.
Installing collected packages: urllib3, idna, chardet, certifi, requests, pywin32, pygments, kivy
Running setup.py install for Kivy-Garden ... done
Successfully installed Kivy-Garden-0.1.4 certifi-2020.12.5 chardet-4.0.0 docutils-0.16 idna-2.10 kivy-
pywin32-300 requests-2.25.1 urllib3-1.26.3
C:\Users\RA>
```

۳. پس از نصب kivy باید تست نمایید که آیا کیوی به درستی نصب شده است یا نه. برای این منظور، IDLE پایتون را اجرا کنید تا شکل زیر ظاهر شود:

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> |
Ln: 3 Col: 4
```

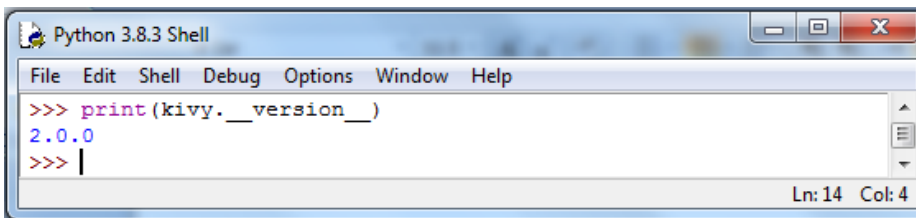
۴. دستور import kivy را مانند شکل زیر اجرا کنید:

```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> import kivy
[INFO ] [Logger      ] Record log in C:\Users\RA\kivy\logs\kivy_21-03-28_3.txt
[INFO ] [deps         ] Successfully imported "kivy_deps.angle" 0.3.0
[INFO ] [deps         ] Successfully imported "kivy_deps.glew" 0.3.0
[INFO ] [deps         ] Successfully imported "kivy_deps.sdl2" 0.3.1
[INFO ] [Kivy         ] v2.0.0
[INFO ] [Kivy         ] Installed at "C:\Users\RA\AppData\Local\Programs\Python\Python38-32\lib\site-packages\kivy\_init_.py"
[INFO ] [Python       ] v3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)]
[INFO ] [Python       ] Interpreter at "C:\Users\RA\AppData\Local\Programs\Python\Python38-32\pythonw.exe"
>>> |
Ln: 12 Col: 4
```

آشنایی با کتابخانه کیوی ۱۹

این شکل نشان می‌دهد که kivy به درستی نصب شده است.

۵. برای تعیین نسخه kivy نصب شده بر روی سیستم، دستور `print(kivy.__version__)` را مانند شکل زیر اجرا کنید:



```
Python 3.8.3 Shell
File Edit Shell Debug Options Window Help
>>> print(kivy.__version__)
2.0.0
>>> |
Ln: 14 Col: 4
```

همان‌طور که در این خروجی مشاهده می‌شود، نسخه kivy نصب شده 2.0.0 است.

۱۳-۱. نوشتن اولین برنامه با Kivy

در این بخش قصد داریم اولین برنامه خود را با Kivy توسعه دهیم. جهت تایپ و ویرایش برنامه نیاز به یک ویراستار داریم. بدون هیچ مشکلی می‌توانید از ویراستار دلخواه خود برای پایتون استفاده کنید. در اینجا، از IDLE پایتون استفاده می‌کنیم. در ادامه برنامه‌ها kivy را در آن تایپ کرده و اجرا می‌کنیم.

مثال ۱-۱. برنامه‌ای که یک پنجره خالی را ایجاد می‌کند.

مراحل طراحی و اجرا

۱. یک فایل جدید پایتون را ایجاد کرده، آن را با نام `ch1-1.py` ذخیره کنید.

۲. بسته kivy را به برنامه وارد کنید.

```
1 import kivy
```

۳. بسته App را به برنامه وارد کنید.

```
1 from kivy.app import App
```

۴. نسخه موردنیاز kivy را تعیین کنید.

```
1 kivy.require('2.0.0')
```

این دستور بسیار مفید است. زیرا از برخورد نسخه‌ها جلوگیری می‌کند. اگر نسخه قدیمی Kivy دارید، باید حداقل نسخه موردنیاز را به‌روز کنید. این دستور اختیاری است. یعنی، می‌توانید این دستور را حذف کنید.

۵. کلاس `firstClass` را پیاده‌سازی نمایید که از کلاس `App` به ارث ببرد.

```
1 class FirstClass(App):
```

۶. متد `build()` را پیاده‌سازی کنید. این متد فعلاً `None` را برمی‌گرداند. چون می‌خواهد پنجره خالی را نشان دهد، بنابراین هیچ ویجتی را برنمی‌گرداند.

۲۰ فصل اول

```
1 def build(self):  
2     return None
```

۷. یک نمونه از کلاس را ایجاد نمایید:

```
1 firstProgram = FirstClass()
```

۸. متد `run()` را بر روی نمونه فراخوانی کنید تا پنجره خالی را نمایش دهد.

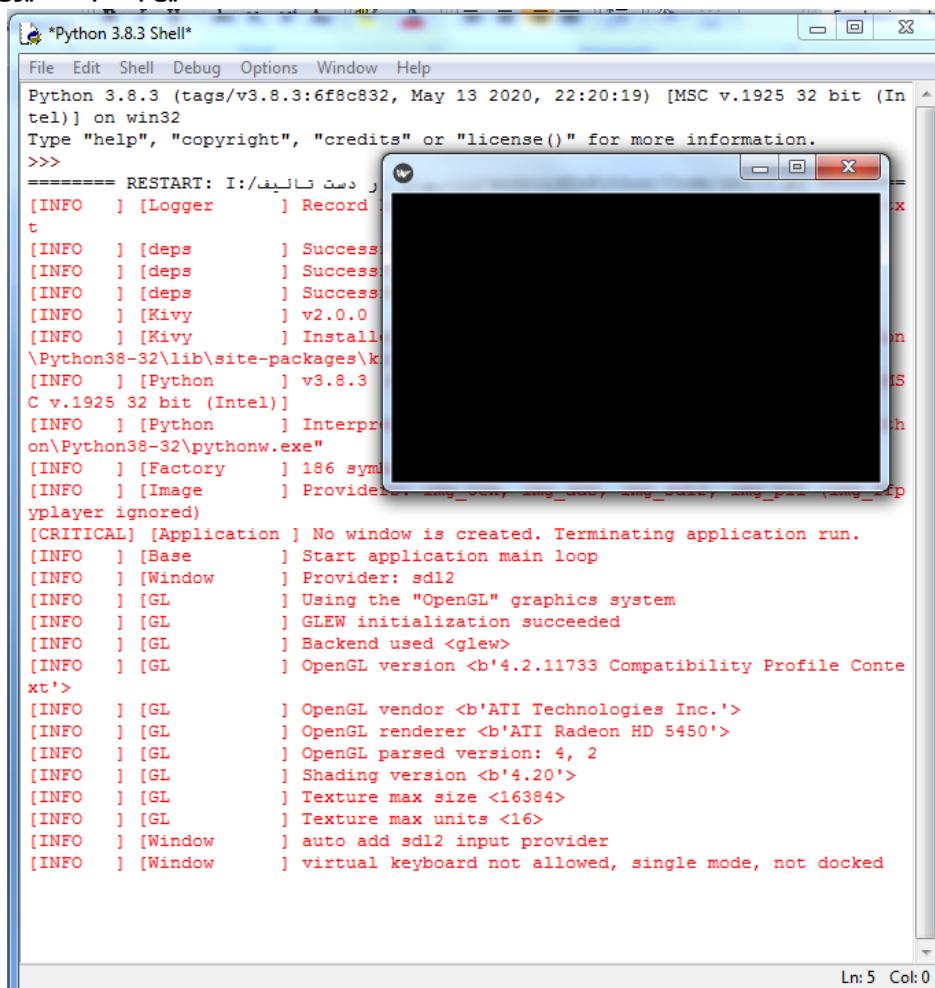
```
1 firstProgram.run()
```

کد کامل این برنامه در زیر آمده ست:

```
1 import kivy  
2 from kivy.app import App  
3 kivy.require('2.0.0')  
4 class FirstClass(App):  
5     def build(self):  
6         return None  
7 firstProgram = FirstClass()  
8 firstProgram.run()
```

۹. برنامه را اجرا کنید تا شکل زیر ظاهر شود:

آشنایی با کتابخانه کیوی ۲۱



```
Python 3.8.3 (tags/v3.8.3:6f8c832, May 13 2020, 22:20:19) [MSC v.1925 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: I:/دست تالیف/
[INFO ] [Logger      ] Record
t
[INFO ] [deps        ] Success
[INFO ] [deps        ] Success
[INFO ] [deps        ] Success
[INFO ] [Kivy        ] v2.0.0
[INFO ] [Kivy        ] Install
\Python38-32\lib\site-packages\k
[INFO ] [Python      ] v3.8.3
C v.1925 32 bit (Intel)]
[INFO ] [Python      ] Interpreter
on\Python38-32\pythonw.exe"
[INFO ] [Factory     ] 186 sym
[INFO ] [Image       ] Provider
yplayer ignored)
[CRITICAL] [Application] No window is created. Terminating application run.
[INFO ] [Base        ] Start application main loop
[INFO ] [Window      ] Provider: sdl2
[INFO ] [GL          ] Using the "OpenGL" graphics system
[INFO ] [GL          ] GLEW initialization succeeded
[INFO ] [GL          ] Backend used <glew>
[INFO ] [GL          ] OpenGL version <b'4.2.11733 Compatibility Profile Conte
xt'>
[INFO ] [GL          ] OpenGL vendor <b'ATI Technologies Inc.'>
[INFO ] [GL          ] OpenGL renderer <b'ATI Radeon HD 5450'>
[INFO ] [GL          ] OpenGL parsed version: 4, 2
[INFO ] [GL          ] Shading version <b'4.20'>
[INFO ] [GL          ] Texture max size <16384>
[INFO ] [GL          ] Texture max units <16>
[INFO ] [Window      ] auto add sdl2 input provider
[INFO ] [Window      ] virtual keyboard not allowed, single mode, not docked
Ln: 5 Col: 0
```

همان‌طور که در این خروجی مشاهده می‌شود، یک پنجره خالی ایجاد گردید.

۱۴-۱. معرفی ویجت‌ها

همان‌طور که در برنامه اول مشاهده کردید، این برنامه یک پنجره خالی را ایجاد نموده و هیچ اطلاعاتی را روی آن نشان نداد. برای نمایش و ورود اطلاعات در پنجره‌ها می‌توانید از ویجت‌های مختلف استفاده کنید. ویجت پایه و اساس رابط کاربری گرافیکی در Kivy است. کیوی یک بوم (canvas) را فراهم می‌کند که می‌تواند برای طراحی روی صفحه استفاده شود. رویدادهایی را دریافت می‌کند و به آن‌ها واکنش نشان می‌دهد. کیوی از ویجت‌های مختلفی پشتیبانی می‌کند. در این فصل ویجت‌های Label، Button و TextInput را بررسی خواهیم کرد.

۱-۱۴-۱. ویجت label

این ویجت برای نمایش پیغام و متن به کاربر استفاده می‌شود که از رشته‌های اسکی و یونیکد پشتیبانی می‌کند. ویجت Label در ماژول kivy.uix.label قرار دارد. پس، برای استفاده از Label ابتدا باید دستور زیر را به برنامه اضافه کنید:

```
from kivy.uix.label import Label
```

یکی از خواص مهم این ویجت خاصیت text است که متن یا عنوان ویجت را تعیین می‌نماید. به عنوان مثال، دستور زیر متن label را عبارت "Hello kivy" تعیین می‌کند:

```
Label(text = 'Hello kivy')
```

مثال ۱-۲. برنامه‌ای که Hello Kivy را نمایش می‌دهد.

مراحل طراحی و اجرا

۱. یک فایل جدید پایتون را ایجاد کرده، آن را با نام ch1-2.py ذخیره کنید.

۲. بسته kivy را به برنامه وارد کنید.

```
1 import kivy
```

۳. بسته App را به برنامه وارد کنید.

```
1 from kivy.app import App
```

۴. ویجت Label را به برنامه وارد کنید.

```
1 from kivy.uix.label import Label
```

۵. نسخه موردنیاز kivy را تعیین کنید.

```
1 kivy.require('2.0.0')
```

۶. کلاس Ch1_2Class را پیاده‌سازی نمایید که از کلاس App به ارث ببرد.

```
1 class Ch1_2Class(App):
```

۷. متد build() را پیاده‌سازی کنید. این متد یک برجسب با عنوان Hello Kivy را برمی‌گرداند.

```
1 def build(self):
2     return Label(text = "Hello Kivy")
```

۸. یک نمونه از کلاس را ایجاد نمایید.

```
1 win = Ch1_2Class()
```

۹. متد run() را بر روی نمونه فراخوانی کنید تا پنجره را نمایش دهد.

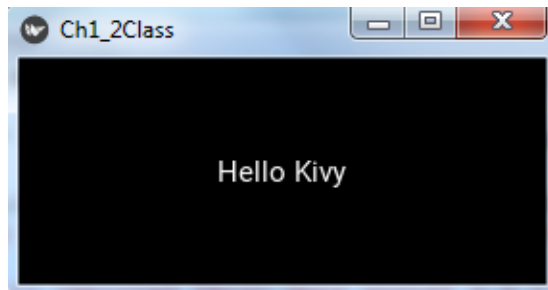
```
1 win.run()
```

کد کامل این برنامه در زیر آمده است:

آشنایی با کتابخانه کیوی ۲۳

```
1 import kivy
2 from kivy.app import App
3 from kivy.uix.label import Label
4 kivy.require('2.0.0')
5 class Ch1_2Class(App):
6     def build(self):
7         return Label(text="Hello Kivy")
8 win = Ch1_2Class()
9 win.run()
```

۱۰. برنامه را اجرا کنید تا شکل زیر ظاهر شود:



همان‌طور که در این خروجی مشاهده می‌شود، پیغام Hello Kivy نمایش داده شده است.

تغییر سبک نمایش متن

سبک متن (سبک مقدار ویژگی text) را می‌توان تغییر داد. به‌عنوان مثال، متن را می‌توان به‌صورت پررنگ (Bold)، زیرخط دار (Underline) و کج (Italic) نمایش داد. برای فعال کردن این قابلیت باید ویژگی markup ویجت را برابر True قرار دهیم و قسمتی از متن که می‌خواهیم به سبک موردنظر نمایش داده شود، مثل Html درون دو تگ قرار دهیم. در زبان Html تگ‌های شروع و پایان را درون یک [] قرار می‌دهیم. اما در Kivy باید هر تگ را درون [] قرار دهیم.

❖ برای Bold کردن از تگ شروع [b] و تگ پایان [/b] استفاده می‌شود. به‌عنوان مثال، دستور زیر kivy را پررنگ نمایش می‌دهد:

```
Label(text = 'Hello [b]kivy[/b]', markup = True)
```

البته برای این منظور همچنین می‌توانید ویژگی bold ویجت را برابر True قرار دهید.

❖ برای Italic کردن از تگ شروع [i] و تگ پایان [/i] استفاده می‌شود. به‌عنوان مثال، دستور زیر کلمه Hello را کج و kivy را پررنگ نمایش می‌دهد:

```
Label(text = '[i]Hello[/i] [b]kivy[/b]', markup = True)
```

البته برای این منظور نیز می‌توانید ویژگی italic ویجت را برابر True قرار دهید.

۲۴ فصل اول

❖ برای رسم خط در زیر متن از تگ شروع [u] و تگ پایان [/u] استفاده می‌شود. به عنوان مثال، دستور زیر عبارت Hello kivy را زیر خط دار نمایش می‌دهد:

```
Label(text = '[u>Hello kivy[/u]', markup = True)
```

البته برای این منظور همچنین می‌توانید ویژگی underline ویجت را برابر True قرار دهید.

❖ برای تغییر فونت متن از تگ شروع [font] و تگ پایان [/font] استفاده می‌شود. مانند دستور زیر:

```
[font = /usr/share/fonts/IranNastaliq.ttf]world[/font]
```

برای تغییر زبان تایپ متن از تگ شروع [text_language] و تگ پایان [/text_language] استفاده می‌شود. مانند دستور زیر:

```
[text_language = en]Kivy[/text_language]
```

❖ برای تغییر اندازه متن از تگ شروع [size] و تگ پایان [/size] استفاده می‌شود. مثل دستور زیر:

```
[size = 20]Kivy[/size]
```

البته برای این منظور نیز می‌توانید ویژگی font_size ویجت را برابر اندازه فونت قرار دهید.

❖ برای تغییر رنگ نوشته متن از تگ شروع [color] و تگ پایان [/color] استفاده می‌شود. مثل

دستور زیر:

```
[color=#ff0000]Fanavarienovin[/color]
```

مثال ۳-۱. برنامه‌ای که Fanavarienovin Publisher را دو سطر با رنگ‌های مختلف نمایش می‌دهد.

مراحل طراحی و اجرا

۱. یک فایل جدید پایتون را ایجاد کرده، آن را با نام ch1-3.py ذخیره کنید.

۲. بسته kivy را به برنامه وارد کنید.

```
1 import kivy
```

۳. بسته App را به برنامه وارد کنید.

```
1 from kivy.app import App
```

۴. ویجت Label را به برنامه وارد کنید.

```
1 from kivy.ui.label import Label
```

۵. نسخه مورد نیاز kivy را تعیین کنید.

```
1 kivy.require('2.0.0')
```

۶. کلاس Ch1_3App را پیاده‌سازی نمایید که از کلاس App به ارث ببرد.

```
1 class Ch1_3App(App):
```

۷. متد build() را پیاده‌سازی کنید.

آشنایی با کتابخانه کیوی ۲۵

```
1 def build(self):
2     labelWidget = Label(
3         font_size = 50,
4         italic = True,
5         bold = True,
6         markup = True)
7     labelWidget.text = '[color=#ff0000]Fanavarienovin[/color]
8                        [color=#00ff00]\nPublisher![/color]
9     return labelWidget
```

سطر اول، متد `build()` را تعریف می کند، سطر دوم، متغیر `labelWidget` را با نوع `Label` تعریف می نماید، سطر سوم، اندازه فونت نوشته را ۵۰ در نظر می گیرد، سطر چهارم، فونت نوشته `label` را ایتالیک (کج) در نظر می گیرد، سطر پنجم، فونت نوشته `label` را پررنگ انتخاب می کند، سطر ششم، به ما اجازه می دهد تا بتوانیم سبک متن را تغییر دهیم، سطر هفتم، نوشته متن را به `Fanavarienovin` با رنگ قرمز تغییر می دهد، سطر هشتم، در ادامه خط بعد عبارت `Publisher` را با رنگ سبز به متن `label` اضافه می کند و دستور نهم، متغیر `labelWidget` را برمی گرداند.

۸. متد `run()` را بر روی کلاس فراخوانی کنید تا پنجره را نمایش دهد.

```
1 Ch1_3App().run()
```

کد کامل این برنامه در زیر آمده ست:

```
1 import kivy
2 from kivy.app import App
3 from kivy.uix.label import Label
4 kivy.require('2.0.0')
5 class Ch1_3App(App):
6     def build(self):
7         labelWidget = Label(
8             font_size = 50,
9             italic = True,
10            bold = True,
11            markup = True)
12        labelWidget.text = '[color=#ff0000]Fanavarienovin[/color]
13                           [color=#00ff00]\nPublisher![/color]
14        return labelWidget
15 Ch1_3App().run()
```

۹. برنامه را اجرا کنید تا شکل زیر ظاهر شود:



همان‌طور که در این خروجی مشاهده می‌شود، پیغام Fanavarienovin Publisher با ویژگی‌های مشخص شده نشان داده شده است.

۲-۱۴-۱. ویجت Button

این ویجت یک واسطه کاربری ایجاد می‌کند که کلیک ماوس را می‌پذیرد. با کلیک ماوس توسط کاربر روی این ویجت، فعالیتی را اجرا می‌کند. ویجت Button در ماژول `kivy.ui.button` قرار دارد. پس، برای استفاده از Button در برنامه ابتدا باید دستور زیر را به برنامه اضافه کنید:

```
from kivy.ui.button import Button
```

بر اساس مستندات سایت کیوی Button از Label مشتق می‌شود. یعنی، دکمه (Button) همان Label است. با این تفاوت که می‌تواند عملی را انجام دهد. پس تمام تگ‌ها و ویژگی‌هایی که برای ویجت Label استفاده می‌شد را در Button نیز می‌توان استفاده کرد.

مثال ۴-۱. برنامه‌ای که ویجت button با عنوان Click Me را به پنجره اضافه می‌کند.

مراحل طراحی و اجرا

۱. یک فایل جدید پایتون را ایجاد کرده، آن را با نام `ch1-4.py` ذخیره کنید.

۲. بسته `kivy` را به برنامه وارد کنید.

```
1 import kivy
```

۳. بسته `App` را به برنامه وارد کنید.

```
1 from kivy.app import App
```

۴. ویجت `Button` را به برنامه وارد کنید.

```
1 from kivy.ui.button import Button
```

۵. نسخه موردنیاز `kivy` را تعیین کنید.

```
1 kivy.require('2.0.0')
```

۶. کلاس `Ch1_4App` را پیاده‌سازی نمایید که از کلاس `App` به ارث ببرد.

```
1 class Ch1_4App(App):
```

آشنایی با کتابخانه کیوی ۲۷

۷. متد `build()` را پیاده‌سازی کنید. این متد یک دکمه با عنوان `Click Me` را برمی‌گرداند.

```
1 def build(self):  
2     return Button(text="Click Me", font_size = 40)
```

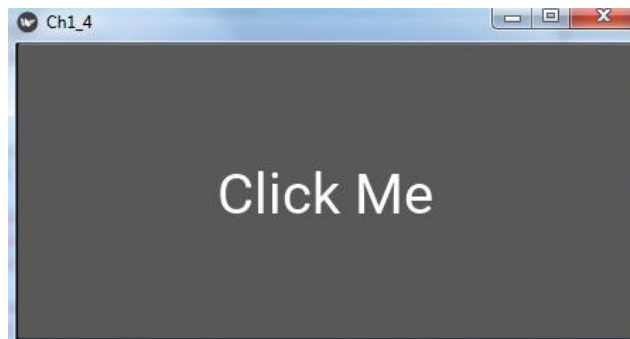
۸. متد `run()` را بر روی کلاس فراخوانی کنید تا پنجره را نمایش دهد.

```
1 ch1_4APP.run()
```

کد کامل این برنامه در زیر آمده است:

```
1 import kivy  
2 from kivy.app import App  
3 from kivy.uix.button import Button  
4 kivy.require("2.0.0")  
5 class Ch1_4App(App):  
6     def build(self):  
7         return Button(text="Click Me", font_size = 40)  
8 Ch1_4App().run()
```

۹. برنامه را اجرا کنید تا شکل زیر ظاهر شود:



همان‌طور که در این خروجی مشاهده می‌شود، یک دکمه با عنوان `Click Me` نمایش داده شده است که با کلیک این دکمه هیچ عملی انجام نخواهد شد.

مثال ۵-۱. برنامه‌ای که ویجت `button` با عنوان `Click Me` و خواص مختلف را به پنجره اضافه می‌کند.

مراحل طراحی و اجرا

۱. یک فایل جدید را ایجاد کرده، آن را با نام `ch1-5.py` ذخیره کنید.

۲. بسته `kivy` را به برنامه وارد کنید.

```
1 import kivy
```

۳. بسته `App` را به برنامه وارد کنید.

```
1 from kivy.app import App
```

۲۸ فصل اول

۴. ویجت Button را به برنامه وارد کنید.

```
1 from kivy.uix.button import Button
```

۵. نسخه موردنیاز kivy را تعیین کنید.

```
1 kivy.require('2.0.0')
```

۶. کلاس Ch1_5App را پیاده‌سازی نمایید که از کلاس App به ارث ببرد.

```
1 class Ch1_5App(App):
```

۷. متد build() را پیاده‌سازی کنید.

```
1 def build(self):
2     btn = Button(text="Click Me",
3                 font_size="20sp",
4                 background_color=(1, 0, 1, 1),
5                 color=(1, 1, 1, 1),
6                 size=(32, 32),
7                 size_hint=(.2, .2),
8                 pos=(300, 250))
9     return btn
```

سطر اول، متد build() را تعریف می‌کند، سطر دوم، متغیر btn را با نوع Button و عنوان Click Me تعریف می‌نماید، سطر سوم، اندازه فونت نوشته btn را ۲۰ در نظر می‌گیرد، سطر چهارم، رنگ زمینه دکمه را نارنجی در نظر می‌گیرد، سطر پنجم، رنگ نوشته دکمه را سفید انتخاب می‌کند، سطر ششم، اندازه دکمه را ۳۲*۳۲ در نظر می‌گیرد، سطر هفتم، مقادیر خواص size_hint_x و size_hint_y را ۰٫۲ انتخاب می‌کند، سطر هشتم، مکان نمایش دکمه را (۲۰۰، ۳۰۰) در نظر می‌گیرد و دستور نهم، ویجت btn را برمی‌گرداند.

۸. نمونه‌ای از کلاس Ch1_5App با نام win را ایجاد کنید.

```
1 win = Ch1_5App()
```

۹. متد run() را بر روی نمونه win فراخوانی کنید تا پنجره را نمایش دهد.

```
1 win.run()
```

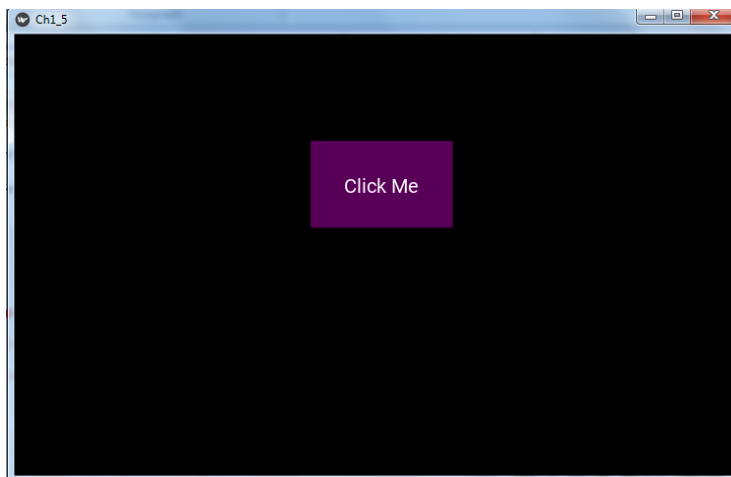
کد کامل این برنامه در زیر آمده است:

```
1 import kivy
2 kivy.require("2.0.0")
3 from kivy.app import App
4 from kivy.uix.button import Button
5 class Ch1_5App(App):
6     def build(self):
7         btn = Button(text="Click Me",
8                     font_size="20sp",
9                     background_color=(1, 0, 1, 1),
```

آشنایی با کتابخانه کیوی ۲۹

```
10 color =(1, 1, 1, 1),
11 size =(32, 32),
12 size_hint =(0.2, 0.2),
13 pos =(300, 250))
14 return btn
15 win = Ch1_5App()
16 win.run()
```

۱۰. برنامه را اجرا کنید تا شکل زیر ظاهر شود:



همان‌طور که در این خروجی مشاهده می‌شود، یک دکمه با عنوان Click Me و ویژگی‌های تعیین‌شده نمایش داده شده است.

مفهوم رویدادها (Events)

جهت آشنایی با مفهوم رویداد در دنیای برنامه‌نویسی، اجازه دهید با یک مثال در دنیای واقعی کار خود را آغاز نماییم. فرض کنید، کارمند بخش فروش یک شرکت به‌منظور دریافت اطلاعات از دو روش تلفن و نامه الکترونیکی، استفاده می‌نماید. به صدا درآمدن زنگ تلفن نشان‌دهنده تحقق یک رویداد بوده که باید به آن پاسخ داده شود. کارمند موردنظر، رخ دادن رویداد را تشخیص و با برداشتن گوشی تلفن و پس از آگاهی از نوع درخواست، عکس‌العمل لازم را انجام می‌دهد (آگاهی از درخواست متقاضی، ثبت سفارش، ارسال کالای موردنیاز).

کارمند موردنظر، می‌تواند از طریق Email نیز اخباری را نظیر سیاست‌های فروش از مدیر خود، دریافت نماید. مثلاً مدیر بخش مربوطه می‌تواند با ارسال یک Email به وی اعلام نماید که قیمت یک کالای خاص افزایش یافته است. پس از ایجاد رویداد فوق، کارمند موردنظر می‌تواند واکنش لازم به‌منظور برخورد با رویداد را با هر روشی