Setting Up Android Studio

Setting up Android Studio is quite straightforward if a little lengthy. Grab some refreshments and get started with the following steps. This tutorial will install Android Studio to the D drive. I chose the D drive because it is a big install, around 12 GB once we have everything downloaded, and the D drive on many PCs is typically larger and has more free space than the

Download Android Studio

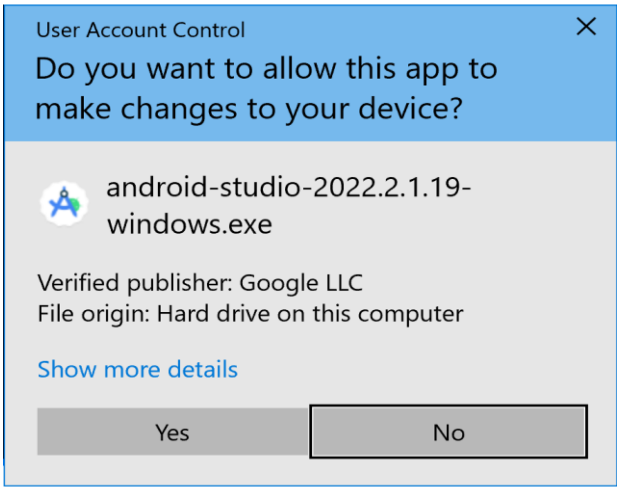
1. Open any web browser and navigate to the [Android Studio download page](#).

This is the Android Developers website, where you can download Android Studio. This page automatically detects your operating system.

2. Click **Download Android Studio**. The **Terms and Conditions** page with the Android Studio **License Agreement** opens.
3. Read the **License Agreement**.
4. At the bottom of the page, if you agree with the terms and conditions, select the **I have read and agree with the above terms and conditions** checkbox.
5. Click **Download Android Studio** to start the download.
6. When prompted, save the file to a location where you can easily locate it, such as the Downloads folder.
7. Wait for the download to complete. This may take a while and may be a good moment to enjoy some tea!

Install Android Studio on Windows

1. Open the folder where you downloaded and saved the Android Studio installation file.
2. Double-click the downloaded file.
3. If you see a **User Account Control** dialog about allowing the installation to make changes to your computer, click **Yes** to confirm the installation.



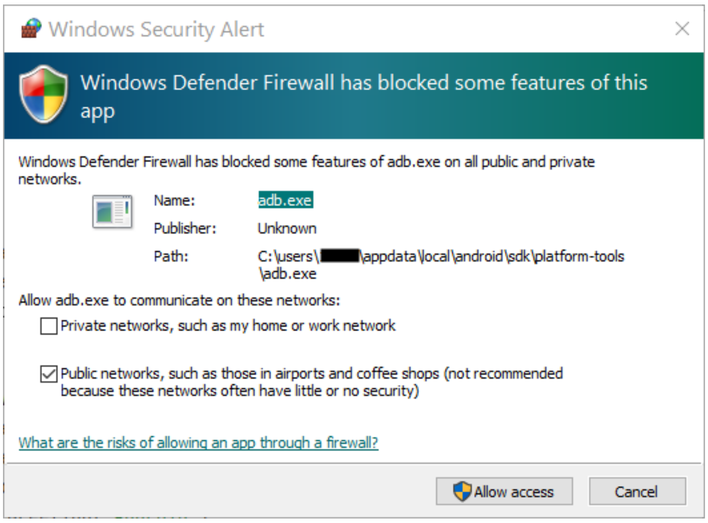
User Account Control

Do you want to allow this app to make changes to your device?

android-studio-2022.2.1.19-windows.exe

Verified publisher: Google LLC
File origin: Hard drive on this computer

4. You may also receive a **Windows Security Alert** about adb.exe. Click **Allow Access,** if needed, to continue the installation.



Windows Security Alert

Windows Defender Firewall has blocked some features of this app

Windows Defender Firewall has blocked some features of adb.exe on all public and private networks.

Name: adb.exe
Publisher: Unknown
Path: C:\users\[redacted]\appdata\local\android\sdk\platform-tools\adb.exe

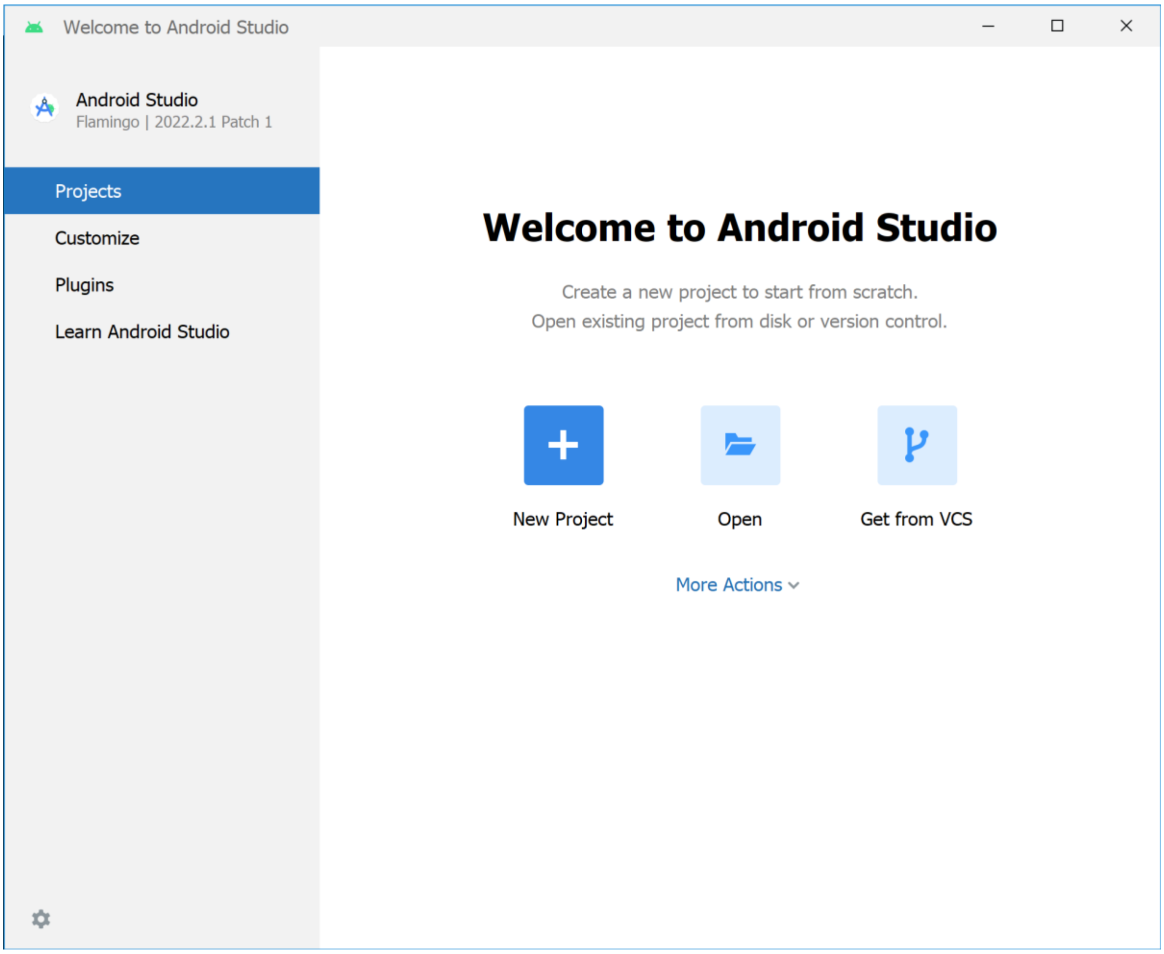
Allow adb.exe to communicate on these networks:

□ Private networks, such as my home or work network

☑ Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)

10. When the download and installation completes, click **Finish**.

The Welcome to Android Studio window displays and you're ready to start creating apps!



Welcome to Android Studio

Android Studio
Flamingo | 2022.2.1 Patch 1

Welcome to Android Studio

Create a new project to start from scratch.
Open existing project from disk or version control.

Introduction to Android

Contents:

What is Android?

Android, developed by Google, serves as an operating system and programming platform designed for a wide range of mobile devices, including smartphones and tablets. It is adaptable to numerous devices from various manufacturers. Android encompasses a software development kit for the creation of original code and the assembly of software modules to produce apps tailored for Android users. Furthermore, it offers a marketplace for app distribution. Collectively, Android forms a thriving ecosystem for mobile applications.

Why develop apps for Android?

Apps are developed for a myriad of purposes, from addressing business needs and introducing new services to creating fresh businesses and delivering games and various content to users. Android is the platform of choice for developers seeking to reach the widest audience of mobile device users. As the world's most popular mobile platform, Android fuels hundreds of millions of mobile devices across more than 190 countries. It boasts the largest installed user base and continues to grow rapidly, with one million new Android device activations daily, each user seeking apps, games, and digital content.

Android provides a touch-screen user interface (UI) for app interaction, primarily relying on direct manipulation through touch gestures such as swiping, tapping, and pinching to interact with on-screen elements. Besides the traditional keyboard, Android features a customizable virtual keyboard for text input. Moreover, Android accommodates game controllers and supports full-size physical keyboards connected via Bluetooth or USB.

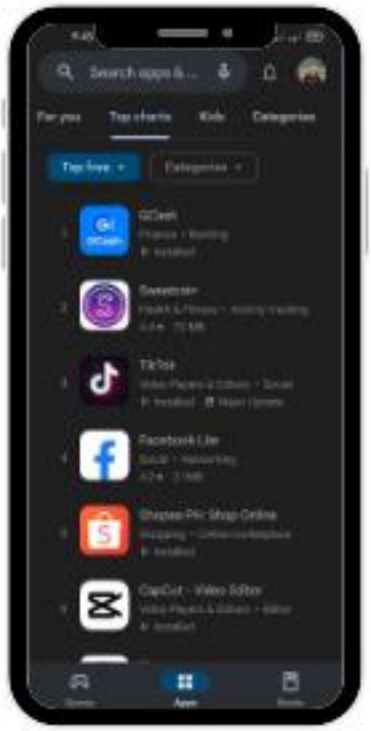
Best experience for app users

The Android home screen hosts multiple pages, featuring app icons for launching associated apps and widgets displaying live, auto-updating content like weather updates, email notifications, or news tickers. Additionally, Android supports multimedia playback, including music, animations, and videos.

The home screen comprises several swipeable pages, enabling users to navigate between them effortlessly. Android is engineered for responsive user interaction, featuring a fluid touch interface and haptic feedback through vibration. Furthermore, many apps leverage internal hardware components such as accelerometers, gyroscopes, and proximity sensors to respond to various user actions. These sensors enable functionalities like automatic screen rotation from portrait to landscape for a broader view or even simulating steering in a racing game by tilting the device like a steering wheel.

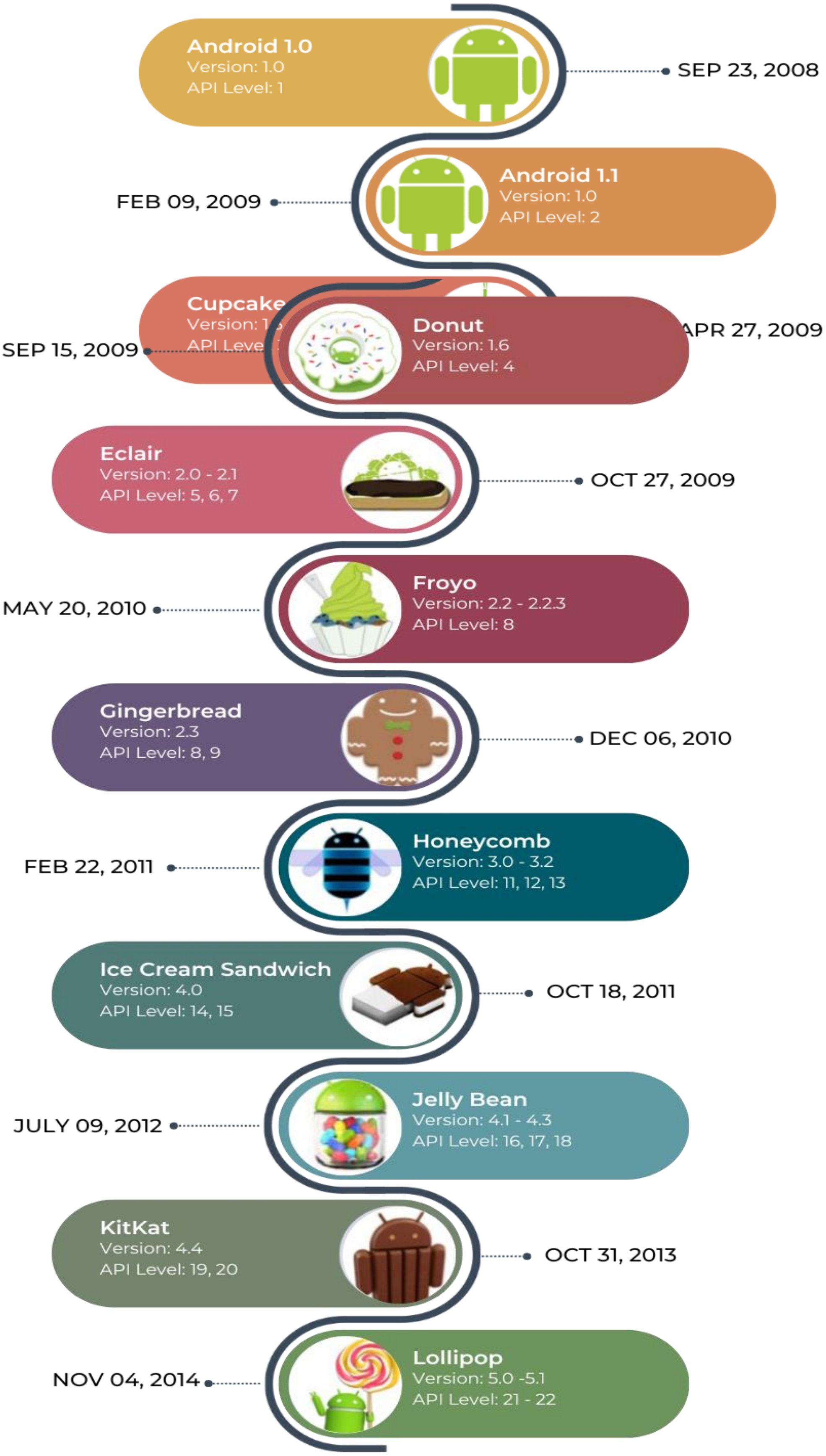
Many distribution options

There are various methods to distribute your Android app, including through email, your website, or by using app marketplaces such as Google Play. The Google Play store, depicted in the figure below, witnesses billions of app and game downloads from Android users each month. Google Play, an official app store for Android, is a digital distribution service established and maintained by Google. It enables consumers to explore and download apps developed using the Android SDK and published through Google.



Android version

Google provides major incremental upgrades to the Android operating system every six to nine months, using confectionery-themed names. The latest major release is Android 13 "Tiramisu".



Android 1.0
Version: 1.0
API Level: 1
SEP 23, 2008

Android 1.1
Version: 1.0
API Level: 2
FEB 09, 2009

Cupcake
Version: 1
API Leve
APR 27, 2009

Donut
Version: 1.6
API Level: 4
SEP 15, 2009

Eclair
Version: 2.0 - 2.1
API Level: 5, 6, 7
OCT 27, 2009

Froyo
Version: 2.2 - 2.2.3
API Level: 8
MAY 20, 2010

Gingerbread
Version: 2.3
API Level: 8, 9
DEC 06, 2010

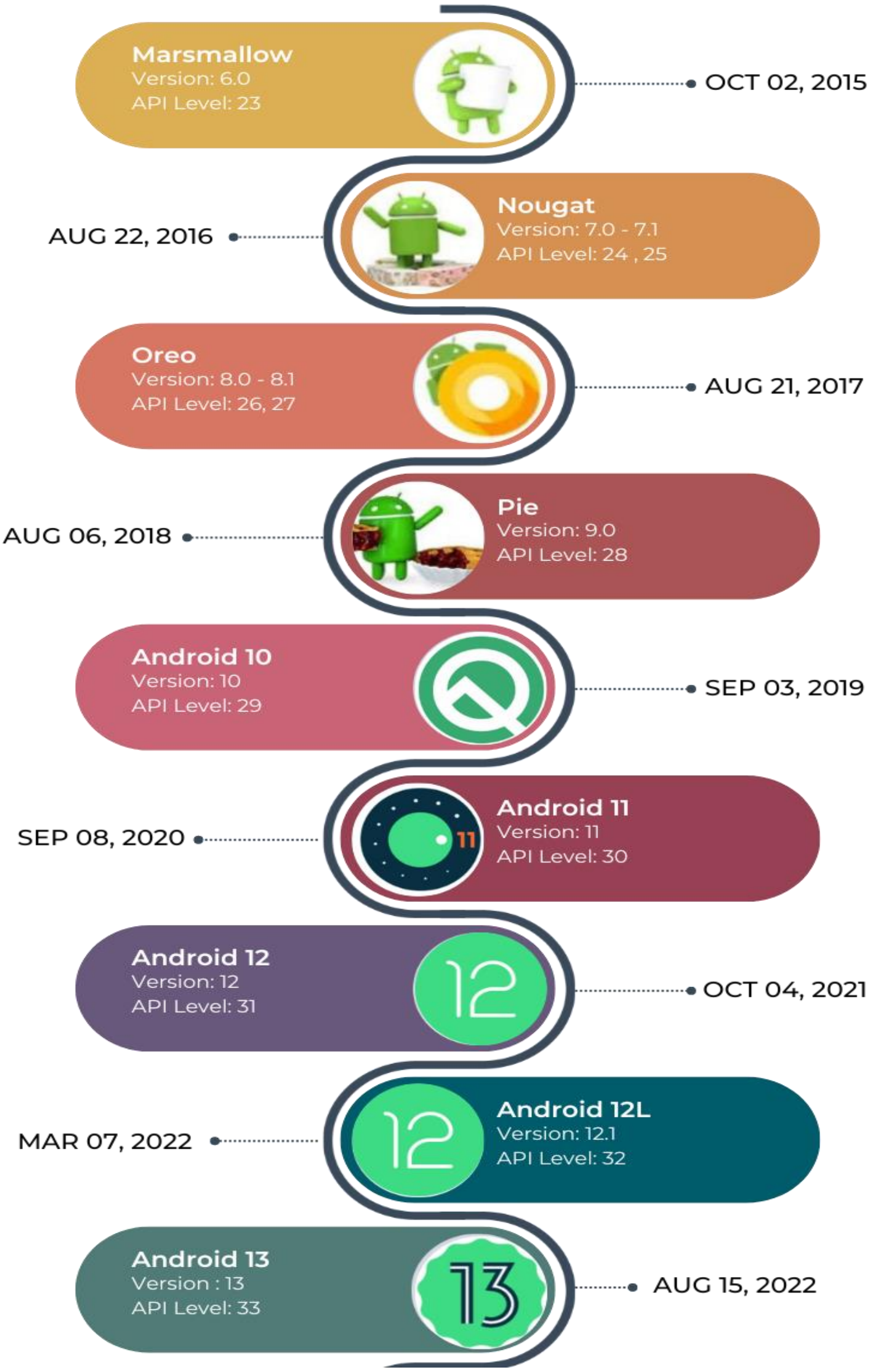
Honeycomb
Version: 3.0 - 3.2
API Level: 11, 12, 13
FEB 22, 2011

Ice Cream Sandwich
Version: 4.0
API Level: 14, 15
OCT 18, 2011

Jelly Bean
Version: 4.1 - 4.3
API Level: 16, 17, 18
JULY 09, 2012

KitKat
Version: 4.4
API Level: 19, 20
OCT 31, 2013

Lollipop
Version: 5.0 -5.1
API Level: 21 - 22
NOV 04, 2014



The Dashboard for Platform Versions is updated regularly to show the distribution of active devices running each version of Android, based on the number of devices that visit the Google Play Store. It's a good practice to support about 90% of the active devices, while targeting your app to the latest version.

The Challenges of Android App Development

While the Android platform provide rich functionality for app development, there are still a number of challenges you need to address, such as:

Building for a multi-screen world

Android powers billions of handheld devices globally, spanning a range of form factors, from wearables to televisions. The multitude of device sizes and shapes directly impacts the design of UI elements in your apps.

Additionally, device manufacturers often introduce their unique UI elements, styles, and color schemes to distinguish their products. These variations extend to features like keyboard layouts, screen sizes, and camera button placements. Consequently, an app may exhibit slight variations in appearance when running on different devices. Or many developers, the challenge lies in crafting UI elements that function seamlessly across this diverse device landscape. Moreover, developers shoulder the responsibility of supplying an app's resources, encompassing icons, logos, graphics, and text styles, to maintain a consistent visual identity across the array of Android devices.

Maximizing app performance

An app's performance, including its speed, network connectivity, and efficient management of battery and memory usage, is influenced by various factors like battery life, multimedia content, and internet connectivity. It's crucial to remain mindful of these limitations and develop your code in a manner that optimally balances and distributes resource usage.

For instance, it's imperative to judiciously manage background services, enabling them only when they are genuinely needed. Such practices not only enhance app performance but also contribute to preserving the user's device's battery life

Keeping your code and your users secure

Securing your code and ensuring a safe user experience with your app is paramount. Employing tools like ProGuard, readily available in Android Studio, can prove invaluable. ProGuard identifies and eliminates unused classes, fields, methods, and attributes, enhancing the efficiency of your code. Additionally, it's essential to encrypt all your app's code and resources during the packaging process to bolster security Sfeguarding sensitive user information, such as login credentials and passwords, necessitates the establishment of secure communication channels. This security must encompass both data in transit (as it travels over the Internet) and data at rest (when stored on the user's device). These measures are fundamental to maintaining the integrity and trustworthiness of your app.

Remaining compatible with older platform versions

When incorporating new Android platform version features into your app, it's imperative to ensure backward compatibility, enabling your app to run smoothly on devices with older platform versions. Prioritizing the latest Android version exclusively is impractical since not all users have the means or willingness to upgrade their devices.