Android studio latest version 32 bit

Launching Android Studio takes just a few clicks. First, be sure to download the latest version of Android Studio. Windows To install Android Studio on Windows, proceed as follows: If you downloaded an .exe file (recommended), double-click to launch it. If you've downloaded a .zip file, open the zipper, copy the Android-Studio folder to your app's files folder, and then open the Bin folder > Android-Studio, and launch Studio64.exe (for 64-bit machines) or studio.exe (for 32-bit machines). Follow the launch wizard on Android Studio and install any group packages that recommend it. Same. The video below shows each step of the setup method when using the recommended .exe download. As new tools and other APIs become available, Android Studio tells you with pop-ups, or you can check for updates by clicking Help > Check for updates. Mac to install Android Studio on your Mac, continue as follows: Launch Android Studio DMG file. Drag and drop Android Studio into the Applications folder, and then launch Android Studio. Choose whether or not you want to enter previous Android Studio settings, then click OK. Android Studio Launch Wizard will guide you through the rest of the setup, which includes downloading android components that are expected to be required for development. Same. The video below shows each step of the recommended setup method. As new tools and other APIs become available, Android Studio tells you with pop-ups, or you can check for updates by clicking Android Studio > Check for updates. Note: If you're using Android Studio on macOS Mojave or later, you might see a notification that allows IDE access to your calendar, contacts or photos. It is quickly created by new privacy protection mechanisms for applications that access files under home directories. So, if your project includes files and libraries in your home directory, and you see this notification, you can choose Don't Allow. To install Android Studio on Linux, continue as follows: Open the .zip file you downloaded in a convenient location for your applications, such as within /usr/local/ for your user profile, or /opt/ for shared users. If you're using the 64-bit version of Linux, make sure you install the required libraries for 64-bit machines first. To launch Android Studio, open a terminal, navigate to Android Studio/Bin/Directory, and run studio.sh. Choose whether or not you want to enter previous Android Studio settings, then click OK. Android Studio Launch Wizard will guide you through the rest of the setup, which includes downloading android components that are expected to be required for development. Tip: To make Android Studio available in your application list, select Tools > Create Desktop Login from the Android Studio menu bar. Libraries needed for 64-bit machines if you are A 64-bit version of Ubuntu, you need to install some 32-bit libraries with the following command: sudo apt- Get installed libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 libbz2-1.0:i386 libbz2-1.0:i386 You are running 64-bit Fedora, the command is: sudo yum install zlib.i686 ncurses-libs.i686 bzip2-libs.i686 which it is. The video below shows each step of the recommended setup method. As new tools and other APIs become available, Android Studio tells you with pop-ups, or you can check for updates by clicking Help > Check for updates. Chrome OS follow these steps to install Android Studio on Chrome OS: If you haven't already, install Linux for Chrome OS. Open the Files app and find the DEB package you downloaded in the Downloads folder below my files. Right-click the DEB package and select Install with Linux (beta). If you've already installed Android Studio, choose whether you want to enter the previous Android Studio settings, then click OK. Android Studio Launch Wizard will guide you through the rest of the setup, which includes downloading android components that are expected to be required for development. Once installed is complete, launch Android Studio either from Launcher, or from chrome OS Linux terminal by running studio.sh in the default installation directory: /opt/android-studio/bin/studio.sh the same. As new tools and other APIs become available, Android Studio tells you with pop-ups, or you can check for updates by clicking Help > Check for updates. Note: Android Studio on Chrome OS now supports deploying your app only to a connected hardware device. For more information, read run apps on a hardware device. World-class code editing, debugging, performance tools, a flexible build system, and an instant build/deployment system all allow you to focus on building unique, high-quality applications. Instantly run push code and resource changes in your app running on a device or emulator and see changes come to life instantly. Run instantly dramatically speed up your editing, build, and run cycles, keeping you flowing. Smart code editor writes better code, works faster, and more productive with smart code editor that helps you every step of the way. Android Studio is built on IntelliJ and is capable of completing advanced code, refactoring, and code analysis. The quick and rich emulator features installing and running your apps faster than with a physical device and testing your app on practically any Android device configuration: Android phones, Android tablets, Android wear, and Android TV devices. The new Android 2.0 emulator is faster than ever and allows you to dynamically resize the emulator and access the set of sensor controls. The robust and flexible build system easily configures your project to include code libraries and generate multiple build species from a single project. With Gredel, Android Studio offers high-performance build automation, strong dependency management, and customizable build configuration. What's new: This minor update includes various bug fixes, as well as support for new default settings for closed visibility on Android 11. Learn more, look at the release notes of the Android Gradle plugin During these uncertain times, we are humbled by many developers around the world who are finding ways to keep doing what they do best to create amazing apps for Android. Whether you're working from your kitchen table on a laptop or from a home office, you need tools that keep up with you. Android Studio 4.0 will depend on our drive result to give you new and improved tools for smarter coding, faster build, and design of your users' apps, and it is now available on stable channels. Some of the highlights of Android Studio 4.0 include a new motion editor to help bring your apps to life, build analysis to check the causes for slower build times, and java 8 language API you can use regardless of the minimum API level of your app. Based on your feedback, we also overhaul the cpu interface profiles to provide more intuitive workflow and easier side analysis of subject activities. And the Improved Plan Inspector now provides live data from your app's UI, so you can easily debugging exactly what is shown on the device. As always, this release will not be possible without initial feedback from our preview users. So in reading or watching below for more highlights and new features you can find in this stable version. If you're ready to jump in and see for yourself, head to the official website to download Android Studio 4.0 now. Motion Editor The MotionLayout API expands ConstraintLayout-rich capabilities to help Android developers manage complex motion and widget animation in their apps. In Android Studio 4.0, this API is made easier with the new Motion Editor—a powerful interface for creating, editing, and previewing MotionLayout animations. You no longer have to create and change complex XML files; Motion Editor generates them for you, with support for set editing limitations, transitions, keyframes, and viewing features. And if you want to see the code the editor creates, it's a click away. And equally easily, for developers already using ConstraintLayout, IDE can easily convert those to MotionLayout. Upgrade Inspector Plan did ever want to check where a value came from for a specific feature? Or view a live 3D representation of the tattoo views to easily inspect your comment hierarchy? With the new plan inspector, debugging your UI is much more intuitive by giving access to data that remains updated with your running program and providing insight into how resources are being resolved. Use the Live Plan Inspector from the main menu by selecting View > Windows Tool > Layout Inspector. If you're deploying to a device running api 29 levels or higher, you'll have access to additional features, such as a dynamic layout hierarchy that will change updates as display, detailed viewing features that also help you determine how resource values are resolved, and a live 3D model of your running app's user interface. Move and transition between views on your running schedule while always having To debugging your UI to pixel perfection. CPU Profiler UI Upgrades The CPU profiler is designed to provide a rich amount of information about your app's thread activity and trace recordings. So when you gave us feedback on how we can navigate an even more intuitive UI and make data easier to understand, we listened. On Android Studio 4.0, CPU recordings are now separated from the original profiler timeline and organized in groups to allow easier analysis. You can move groups up and down, or drag and drop individual items inside a group for additional customization. For simpler side-by-side analysis, you can now view all subject activities in the subject activity timeline (including methods, functions, and events) and try new navigation shortcuts to easily move around the data—such as using the W, A, S, and D keys to zoom fine-grained and panning. We also redesigned the UI tracking system so unique events for better visual distinction of color, topics sorted to level the busier ones first, and you can now focus on seeing data for the only topics you choose. Finally, we invested in cpu profile quality, and as a result we saw a significant reduction in the user-reported error rate of recording from Android Studio 3.6. Full release notes can be found here. here.