DOWNLOAD PDF XCODE 9.2 TUTORIAL

Chapter 1 : How to Install Xcode on Windows 10, 8 or and 7 for iOS SDK

In this video, we are going to learn about auto layout in iOS app development. Auto layout allows developer to adjust position of their controls automaticall.

These are called universal binary files, which allow software to run on both PowerPC and Intel -based x86 platforms and that can include both bit and bit code for both architectures. Composition[edit] The main application of the suite is the integrated development environment IDE , also named Xcode. Up to Xcode 4. Starting with Xcode 4. One technology involved was named Shared Workgroup Build, which used the Bonjour protocol to automatically discover systems providing compiler services, and a modified version of the free software product distcc to facilitate the distribution of workloads. Earlier versions of Xcode provided a system named Dedicated Network Builds. These features are absent in the supported versions of Xcode. As of Xcode 3. Xcode 3 still includes the WebObjects frameworks. The next significant release, Xcode 1. It supported shared precompiled headers , unit testing targets, conditional breakpoints, and watchpoints. It also had better dependency analysis. Notable changes since 2. It also supports Project Snapshots, which provide a basic form of version control; Message Bubbles, which show build errors debug values alongside code; and building four-architecture fat binaries 32 and bit Intel and PowerPC. It included the GCC 4. Another new feature since Xcode 3. It supports static program analysis , among other features. It also drops official support for targeting versions earlier than iPhone OS 3. But it is still possible to target older versions, and the simulator supports iPhone OS 2. Also, Java support is "exiled" in 3. Version 4 of the developer tools consolidates the Xcode editing tools and Interface Builder into one application, among other enhancements. The deployment target can still be set to produce binaries for those older platforms, but for Mac OS platforms, one is then limited to creating x86 and x binaries. Later, Xcode was free to the general public. On August 29, , Xcode 4. On October 12, , Xcode 4. On September 19, , iOS 6 and Xcode 4. On January 28, , iOS 6. This latest was only included in the betas version. Apple removed support for building garbage collected Cocoa binaries in Xcode 5. Xcode could be downloaded on the Mac App Store. It introduced support for Swift 3. It introduced support for Swift 4 and Metal 2. Xcode 10 introduced support for the Dark Mode announced for macOS Mojave , the collaboration platforms Bitbucket and GitLab in addition to GitHub , training machine learning models from playgrounds, and the new features in Swift 4.

Chapter 2 : Xcode for Windows? - Thomas Hanning

Please subscribe to our channel and hit the bell icon so that you get notified as soon as we upload new video tutorial on swift 4. (Swift 4 + Xcode) - Duration:

This tutorial has been updated to Xcode 9. The original tutorial was written by Ron Kliffer. With the release of iOS 6 in , Apple made a dramatic change and replaced Google Maps with an in-house mapping engine: There are benefits and drawbacks to both MapKit and the Google Maps iOS SDK, but this tutorial will walk you through implementing Google Maps into your apps to see for yourself how well it can work in your geolocating apps. This tutorial assumes some familiarity with Swift and iOS programming. This tutorial also requires familiarity with CocoaPods. You must have CocoaPods installed in order to follow this tutorial. To learn more about CocoaPods, check out this tutorial by Joshua Greene, published right here on the site. This tutorial uses Xcode 9. Take a look around to get familiar with the project. The important elements to notice are: This is a wrapper class for making Google API calls. This is a model for place results returned from Google. This is a subclass of UIView that displays details of places. It comes with a matching xib file. Press the action button on the right side of the navigation bar to see the TypesTableViewController screen like so: Select the newly created project to access its settings. Search and enable these APIs: Click Create credentials, and then click API key to create the key: Copy the key and click Close: Downloading dependencies Installing GoogleMaps 2. A constant to hold your Google API key. Bring up the Object Library by selecting the third tab in the view toolbar â€" Utilities â€" and then select the third tab in the library toolbar â€" Object Library â€" as shown in the screenshot below: Your MapViewController scene should now look like this: Select Map View in the Document Outline and then choose the second button from the right in the bottom right of the Interface Builder window â€" the Pin button. Ensure that Constrain to margins is unchecked â€" this ensures that the map will fill all the available space on the screen â€" and add 0 zero space constraints from the top, left, bottom and right of the superview. Your Pin editor should look like this: Click on Add 4 Constraints to add the constraints to the map view. Before you build and run the project, add an IBOutlet for the map view. To do that, bring up the Assistant Editor by selecting the second tab in the Editor toolbar: Select the map view in Interface Builder, hold down the Ctrl key and drag a line from the map view to MapViewController. A popup will appear; set the connection type to Outlet and the name to mapView. Before you build and run, add the following to the top of the file, after import UIKit: Next, find viewDidLoad and add these two lines to the bottom: By accessing your location, this app can find you a good place to eat. Add the following extension to the bottom of MapViewController. Here you verify the user has granted you permission while the app is in use. The GMSCameraPosition class aggregates all camera position parameters and passes them to the map for display. You should now see a map centering around your location. Scroll the map and tap the Locate button and the map will center back to your location like so: Google Maps has an object that does exactly that: This takes a simple coordinate and returns a readable street address. Next, open the Attributes inspector, and give the label the following attributes: Set Alignment to center. Set Lines to 0. Surprisingly, this lets the label take up as many lines as it needs to fit the text. Finally, add to the label left, bottom and right constraints of 0 as shown below: This pins the label to the bottom of the screen and stretches it over the entire width of the screen. Your storyboard scene should look like the following: Next, create an outlet for the label. Set the connection type to Outlet, the name to addressLabel and click Connect. This adds a property to your MapViewController that you can use in your code: IBOutlet weak var addressLabel: Add the method below to MapViewController: Creates a GMSGeocoder object to turn a latitude and longitude coordinate into a street address. Asks the geocoder to reverse geocode the coordinate passed to the method. It then verifies there is an address in the response of type GMSAddress. This is a model class for addresses returned by the GMSGeocoder. Sets the text of the addressLabel to the address returned by the geocoder. Add another extension to the bottom of MapViewController. Next, add the following line of code to viewDidLoad: Finally, add the following method to the newly added extension: Notice anything wrong with this picture? When padding is applied to the map, all of the visual elements will be placed according to that padding. Prior to the

animation block, this adds padding to the top and bottom of the map. Build and run your app again; this time the Google logo and locate button will move to their new position once the label becomes visible: It receives a Bool that tells you if the movement originated from a user gesture, such as scrolling the map, or if the movement originated from code. You call the lock on the addressLabel to give it a loading animation. Add the following to the top of reverseGeocodeCoordinate coordinate: For the full implementation of the lock and unlock , check out UIViewExtensions. Build and run your app; as you scroll the map you should see a loading animation on the address label like so: Google Places API is a free web service API you can use to query to find establishment, geographic locations, or other points of interest near any given point. Each marker object holds a coordinate and an icon image and renders on the map when added. Ensure you choose Swift as the language for this file. Replace the contents of PlaceMarker. Add a property of type GooglePlace to the PlaceMarker. Next, add two more properties to MapViewController. Add the following method to MapViewController: Clear the map of all markers. Iterate through the results returned in the completion closure and create a PlaceMarker for each result. This line of code is what tells the map to render the marker. First, the user can reasonably expect to see places nearby when the app launches. Choose Action and name the method refreshPlaces. Insert the following code into the newly added method: Change the search types in the TypesTableViewController and see how the results change: If you return a view, then it pops up above the marker. If nil is returned, nothing happens. How does that happen? You first cast the tapped marker to a PlaceMarker. Next you create a MarkerInfoView from its nib. Then you apply the place name to the nameLabel. If so, add that photo to the info view. If not, add a generic photo instead. Obviously, the pin needs to re-appear at some point. Returning false again indicates that it does not override the default behavior when tapping the button. Scrolling the map closes the infoView and brings the pin back: You now have a fully functioning Google Maps app. Since the Google Maps SDK is very large in size, we excluded it from the project, so make sure to run pod install before building. Uniform experience for cross platform iOS and Android apps.

Chapter 3 : [IOS] Realm Database Tutorial (Xcode & Swift 4) - KodeChamp

In this tutorial we are going to start off with a single view app and we are going to name it TodoList with CoreData. We are going to choose Use Core Data. On the next screen, I am going to select Create Git Repository on MAC because I want to upload the project to Github.

How to Install Xcode on Windows 10, 8 or 8. Xcode is an integrated development environment IDE that consists of set of software development tools which are designed by Apple specifically for developing software on Mac OS X and iOS. There are many reasons and compatibility issues behind this unavailability. If you are quite firm to install Xcode on your Windows PC 7, 8. Xcode is an interface builder which can also be considered as a testing application and an asset management toolkit. Here is a method on how to install Xcode on Windows PC 7, 8 or 8. Before initiating the installation process, you need to have the following system requirements: Downloaded Xcode package from Apple site. You have to create the virtual machine on your oracle virtual box. Now, you will get a new window asking for the name of the new operating system. You need to select the size of RAM for the virtual machine. Select the memory size and then click Next. Now, Select and create the type of virtual hard drive file. Now, you need to allocate the file loaction and size of Android from physical hard drive on your device. Then, Click on Create. Now, you have successfully created virtual machine on your virtual box. You need to mount the iso file which is downloaded before. Sign in using your Apple ID into the App store. You need to enter the Apple ID and password as shown in the image below: It shows you various related apps. Locate Xcode from different apps and click on Free and Download. After completing the installation process, open it from the applications. Enter your name and password and press OK. Now, you can create new projects and better apps using this interface. However, it can be used for educational purpose, but not for professional app developers. So in this way you can easily download and install Xcode on your any windows Personal computer or laptop and you can create better apps as per your need.

Chapter 4 : Playing sound in Xcode |Apple Developer Forums

This Xcode tutorial is updated for Xcode If you're using an earlier version, I'd recommend to update to the latest version because there are significant changes to the Swift programming language that you'll only get with the latest version of Xcode.

The source code editor lets you transform or refactor code more easily, see source control changes alongside the related line, and quickly get details on upstream code differences. You can build your own instrument with custom visualization and data analysis. Swift compiles software more quickly, helps you deliver faster apps, and generates even smaller binaries. Test suites complete many times faster, working with a team is simpler and more secure, and much more. Shine in the Dark Code you write in Xcode looks stunning as the dark Xcode interface makes your work the star of the show. The entire interface is tuned for your dark Mac experience, from icons, to fonts, to the subtle contrast color of the Jump Bar. Xcode also gives you powerful tools for creating your own dark apps for macOS. Interface Builder lets you quickly switch your design and preview from light to dark. Asset catalogs define assets and named colors. And you can switch your app in and out of Dark Mode while debugging. This is all done using controls within Xcode that only apply to your app. No need to change your system settings. In Markdown files, headings, bold and italic text, links, and other formatting are instantly rendered in the editor as you type. Code Like a Pro Xcode includes a lightning-fast source code editor. Text scrolls incredibly smoothly, even when editing enormous source files. Smooth animations are used throughout, whether folding your code to enhance focus, or when Xcode highlights errors and offers Fix-its. With great Markdown support, your accompanying documentation will look great, too. And because the transformation engine is open source as part of swift. Work as a Team Source control is the place where your whole team works on code together. Xcode supports working directly with several collaboration platforms, including: To make your workflows easier and more secure, Xcode can even generate a unique SSH key for you and upload it to the server. Once logged in to your favorite service, the Xcode clone window shows all of your personal and saved repositories. From this window you can also search for additional repositories on your connected servers and quickly check any of them out with just a click. You can even rebase your changes when pulling latest versions. The source control navigator in Xcode makes it easy to view each of your branches, tags, and remotes with a timeline of commits. Inspect an entry to see all affected files or double-click on a commit to see everything that changed. Common operations, such as creating and merging branches, are quickly accessible in the navigator. Customize Your Debugging Tools Using Instruments, you can finally retire your print statements, which are replaced with OSLog signposts and your own custom instruments. With virtually no overhead you can mark important points throughout your code, then track those signposts as your app runs in Instruments. You can go even further and build your own instrument with custom visualization and data analysis. Xcode includes templates so you can build instruments using the same tools that Apple uses. Your instruments can easily be shared as part of your project and installed by other team members or users of your public frameworks. Xcode also collects anonymous energy and crash logs from your users, highlighting the most important issues and letting you dive directly to the offending lines of code. Simulate and Test Xcode includes a robust testing engine built right in. Run unit tests, as well as UI and performance tests, across multiple physical devices at a time. Or take advantage of the processing horsepower of Mac to dramatically speed up testing using simulated devices running in parallel. For continuous integration setups, you can launch many different simulated device types to run your complete test harness from beginning to end. Or, to complete your tests as fast as possible, Xcode can spawn many clones of a single simulated device, fanning out all of your tests to finish in a fraction of the time. You can also dedicate another Mac in your network to host Xcode Server for automated building and testing. Build Swiftly Xcode 10 includes Swift 4. Compared to Swift 4. Optimized for the latest multi-core Mac hardware, Xcode and Swift make for a lightning-fast development platform. In Xcode 10, playgrounds are dramatically enhanced to work more like a traditional REPL, while making the live view even more responsive and fun to use for quick designs. As you add new code, only new lines are recompiled. You can choose to re-run specific

lines of code, or hit shift-return to run the program right up to the line of code you just typed. The new incremental model is a perfect fit for working with the new Create ML framework. Train your models directly within a playground, alongside the code that will use the model in your app. When ready, just drag-and-drop your newly trained model into your app.

Chapter 5 : objective c - Create Command Line Tool with XCode 9+ - Stack Overflow

There are three database options out there for swift developers. SQLite CoreData Realm Core Data claims to be faster than than SQLite but it takes up more space and it uses more memory for storage of contents than SQLite. and Realm claims to be faster than Core Data.

You will become more efficient in programming, just by knowing how your environment works. This is a perfect tutorial if you are a beginner in iOS Development. This is probably the most used area as it contains many key features that Xcode offers called navigators. I would start explaining them one by one from left to right. This navigator enables you to add, remove, edit or group files. Always keep this tab in focus. It has an integrated support for GitHub accounts, which enables you to manage your repositories directly from your sidebar, and push changes to the cloud without having to use other tools. This is only available from Xcode 9. Instead of going through your files to find the method you are looking for, just click on the file you need and then click the desired method or property definition. Useful for files that contain lots of lines of code. You can display the symbols in a hierarchical or flat list. This makes a global search for the given text and returns results that are matching. You can also include various filters in the search. Errors are shown with a red color, while warnings are yellow. It will provide you with a detailed log of what is going on. You can display Buildtime or Runtime issues. It will provide you with exact lines where the app has stopped and provide you with a reason in the console. From here, you can easily set breakpoints and monitor their activities. I really love how clean this feature is. You should create a bot, and it will provide you access to a detailed information about your bots and the integrations performed on the server. Many iOS apps use Google Maps. This is a very common feature, so I have decided to prepare an ultimate guide on Googleâ€¦theappspace. On the right side, you can see another sidebar with various tabs. But, unlike the left sidebar, this one contains a different set of tabs depending on your location. On this side, the tabs are called inspectors. File Inspector - this inspector provides basic details and settings about the selected file. Identity and Type - provides you with an information where the file is located in the directory and gives you a possibility to open it in Finder. Also, you can set the location of the file whether it should be an absolute or relative path. This tells the project that the file belongs to its target. Text Settings - Personally, I have never used this part. You can set up indentation and other text settings from here. These settings are only for the selected file. Quick Help Inspector - provides you with a documentation for a selected class. For example, find a String class inside your file, and move the cursor to that word. A quick explanation about the class will appear. Storyboard File If you click on a. File Inspector â€" this inspector contains the same details that I have explained above, just with a few new sections. Localization - shows all your localization files. Quick Help Inspector - same info as above. By identity we understand, assigning a custom class, providing a storyboard ID so you can access the view controller via code, and User Defined Runtime Attributes where you can add various styling properties, instead of adding them via code example: Attributes Inspector - this inspector is used for adjusting the properties of the selected object. Each object contains its own set of properties. For example, the UILabelcontains settings like adjusting text, text color, font, background color etc. You can also add your own properties, by using IBInspectable. Size Inspector - I think the name explains it all. Anything related to the sizing of the object can be found here. Connections Inspector - used for communication between the code and the view controller using IBOutlet and assigning actions via IBAction. I will go through the. Two of the three tabs are already familiar to you from the previous points above. Attributes Inspector - from here, you can adjust the properties for the selected asset. Most common properties are image compression, rendering type, device support, scale vector or individual etc. Header Next on the list is the header. We are going a little bit above the left and right sidebar, at the very top of Xcode. Right Side A place where you can apply different options to the editor. I would start explaining them from left to right. Standard Editor - this control represents the default view of the editor. Assistant Editor - By using the assistant editor, your coding area will split into two parts. The left side represents your file, and the right side represents only your method definitions. What it does is, it ignores the code written inside your methods, and just shows you the method definitions. Useful if

your class contains lots of code. Version Editor - Splits your coding area and adds a replica of your class which tells you what you have changed from the last commit. This is used for version control, providing you with a comparison of what was changed. The next three icons that are located in the top right corner, represent the showing and hiding of the sidebars and the debug area. It is used if you lack space on your screen i. We will go through the project settings. I am sure you have seen this screen before, so I will start explaining the tabs directly. General General - the name explains it all. Here, you can find settings that are general for the specific target. Identity - controls the app name, bundle identifier, version number and build. Basically, it signs the app for sending versions of the app for testing or production. Deployment Info - from here, you can add minimal iOS version support, decide if the app is going to be Universal or only iPhone or only IPad, and add the default Storyboard file. App Icons and Launch Images - assign app icon and splash screen assets. Capabilities - contains switches with various services that you can use. Resource Tags - you can assign tags to resources, and they will all end up here. I have never used this feature, but if you want to find out more about it, you can check here. Info - this tab shows the properties from your active. For example, the information in a build setting can specify which options Xcode passes to the compiler. From assigning provisioning profiles to adding directory paths for 3rd party libraries. Each target has its own Build Settings. Build Phases - provides you a list of all your files that will be included in the compile. All your frameworks, assets,. Also, there is a Run Script feature where you can add your own shell script. Build Rules - Xcode gives you a possibility to write your own rules. If you got decent scripting skills you can write almost anything. I hope that you enjoyed this tutorial and that it helped you understand more about the basic functionalities in Xcode.

Chapter 6 : Xcode - Wikipedia

My first uploaded tutorial! Great for beginners! This tutorial covers Labels and Buttons and connecting them with code! Thanks for watching!

Video Should you install macOS on a Windows machine? Then, Xcode could be installed. There are many sources in the internet that promises you that this were actually a good idea. An alternative for buying a Mac is renting a Mac in the cloud. There are several providers who offer that service, for example this one. In this particular case, Xcode is already installed, so that you can start out immediately. Non-native ways for developing iOS apps Many people, who wish to install Xcode on a Windows machine, wants to develop iOS applications. There are options to develop iOS apps without using Xcode directly. This are apps, that just have a bunch of HTML5 sites, that gets displayed in an internal web view. In theory, this makes it also possible to run the apps not only on iOS, but on Android devices as well. Develop once, deploy twice? Most importantly, those apps feel very bad. The users are used to have native apps, and non-native apps always feel very cumbersome. The same holds true for all other types of cross-platform development systems as well. And for that you need a Mac. In this case you can start out by learning Swift. Before Swift, those application were written by using Objective-C. Swift is open source and there are several ways for writing Swift code on a non Mac machine. First, there is the Swift Playgrounds app for the iPad. You can use it to write Swift code and also to learn how to do it. By the way, many people wish there were an Xcode version for the iPad. You never know what the future brings, but at the time of this writing there are no hints that this will change anytime soon. Besides that, there are websites, that allow you to write and run Swift code, for example here. Since Swift is open source, there are even projects that allow you to use Swift for programming Windows applications. For example, you can take a look here. The support for Linux is even better and you can download it directly from the Swift website. If you want to learn more about Swift, take a look at the Swift category on my blog. There are also a lot of other good tutorials out there. At the time of this writing, the newest Xcode version 9. There are nice refurbished devices in the Apple Store, but used devices are also a good alternative. Conclusion Unfortunately, there is no Xcode for Windows. And there is also no good and legal way to install macOS on a Windows machine. Renting a MacOS machine in the cloud is an alternative: And Swift compilers are even available for Windows and Linux.

Chapter 7 : Lazy Foo' Productions - Setting up SDL 2 on iOS with XCode

This video uses the latest and greatest from Apple (Xcode 9, Swift 4, iOS 11) and will teach you how to build a really cool iPhone app for iOS This tutorial is designed for absolute beginners.

Core data is not a database. It is a persistence framework and it is performant. Alternatives include Realm and Firebase which are very good if you plan to build cross platform apps. Creating the Project In this tutorial we are going to start off with a single view app and we are going to name it TodoList with CoreData. We are going to choose Use Core Data. Creating the Model The first thing that we will do is work on our object graph. You may have to scroll down to see Core Data. You can name the model what ever you like or you can leave it as the default model. We are going to select our data from the navigator and click on add entity. An entity is like a class in swift. The entity can be customized by giving it a name and attributes. Think of the attributes as being similar to the properties in a class. We are going to give this entity two attributes. The first one is going to be called name. The type is going to be a string and we must deselect the box making it optional it is selected in the picture. The second attribute will be called completed and we will make it a boolean and we can set the default value to NO. Just like before we need to make sure that the optional checkbox is not selected. In the attributes inspector set it to the initial view controller. Rename the class to ToDoViewController. Your code should look like this:

Chapter 8 : [IOS] TodoList with Core Data (Xcode and Swift 4) - KodeChamp

1. iOS 11 & Swift 4 - The Complete iOS App Development Bootcamp One of the best tutorials on the subject, more than 58, students have enrolled for this program. You will be taught iOS 11 App Development from scratch using both Xcode 9 and Swift 4.

Chapter 9 : swift - Playing sound in Xcode - Stack Overflow

Introduction to Xcode 9 That's why I would like to dedicate some time around the new Xcode 9. This is a perfect tutorial if you are a beginner in iOS Development.