

Chapter 1 : MTV Original TV Shows, Reality TV Shows | MTV

Part 3 -- Workshop on "The Real World" FndtnACIM. Gloria and Kenneth Wapnick present a workshop on the process of attaining the real world. Kenneth discusses the meaning of "sin" as used in "A."

The concept of corporate social responsibility, that businesses should both self-regulate and benefit their communities, goes back to before the s. Shortly thereafter, in , the Supreme Court ruling that corporations were legal persons citizens protected by the Constitution negated that right. So if being a good corporate citizen is voluntary, and presumably costly, why would any corporation choose to do so? The answer is a fairly simple one: And it is paying off. But how are they doing it? How are they benefiting? And how can you follow their leads? Just look at these three great examples of corporate social responsibility. For example, Google Green is a corporate effort to use resources efficiently and support renewable power. But recycling and turning off the lights does more for Google than lower costs. Investments in these efforts have real-world effects on the bottom line. Google has seen an overall drop in power requirements for their data centers by an average of 50 percent. These savings can then be redirected to other areas of the business or to investors. Installing energy-efficient lights, allowing telecommuting, and recycling will not only improve your world, it will result in quantifiable cost savings that you can see in the bottom line. Xerox The printing giant offers many programs supporting corporate social responsibility. Their Community Involvement Program encourages it by directly involving employees. Since , more than half a million Xerox employees have participated in the program. The return for Xerox comes not only in community recognition, but also in the commitment employees feel when causes they care for are supported by their employers. By incorporating a limited number of billable hours per year for volunteer efforts, you will enjoy the dual effects of helping your community as well as increasing your employee morale and therefore productivity. Perhaps organizing large group activities for charities such as Habitat for Humanity can bring your group together, and you can spread your name with inexpensive event T-shirts. Sure, mega-corporations can have volunteer programs or philanthropic arms that focus on big-picture issues, but that seems so highbrow. If that is how you think, then take a look at Target. While many shoppers may think of it as just another big-box retailer, Target is more than just a place to buy tires and milk, they are a prime example of corporate social responsibility. Since , Target has been committing more and more effort and assets toward local and environmental support for the communities in which they have stores. By supporting any good cause in your community, you provide two important factors that pay dividends. You have employees who are proud to work for you and clients who are proud to be associated with you. The financial return of either can be many orders of magnitude. Bring These CSR Examples Home Many of these programs, such as allotted time for volunteer work and community donations which Autodesk does with its Autodesk Foundation can begin as small-scale efforts. The cost to the bottom line will quickly be reimbursed when you see your socially active employees producing more and your supported community engaging with your firm. The bonus is that they will get so much in return.

So far in this 3-part series, we have looked at variables (Part 1) and functions (Part 2) in order to leverage them to our advantage. In this third and final part of the real-world DataWeave series, we will look at another common problem area, that of performing nested loops in data structures.

Therefore we also need certain functions to check if employees or locations exist. For the sake of good design, we need at least two Repositories: But a surprising amount of functionality is needed by both of these repositories: They both need to get an object by its ID. They both need to check if an object with a given ID exists. They both need to save a changed object back into the Read Data Store. They both need to be able to get multiple objects of the same type. Consequently, we can build a common IBaseRepository interface and BaseRepository class which implement these common features. The IBaseRepository interface will be inherited by the other repository interfaces; it looks like this: This class provides methods by which items can be retrieved from or saved to the Redis Read Data Store: StringGet key ; if serializedObject. StringGet key ; return! Because of the work we did in the BaseRepository, our Read Model Object repositories will be quite simple. We can think of this as changing when the work of making a collection is done. In SQL Server or other relational databases, most of the time you do the work of creating a collection when you run a query. So, you might have a huge table of, say, vegetables, and then create a query to only give you carrots, or radishes, or whatever. But in Redis, no such querying is possible. Therefore, instead of doing the work when we need the query, we prep the data in advance at the point where it changes. Consequently, the queries are ready for consumption immediately after the corresponding event handlers are done processing. Building the Event Handlers Whenever an event is issued by our system we can use an Event Handler to do something with that event. In our case, we need our Event Handlers to update our Redis data store. In any case, notice that all this event handler is doing is creating the corresponding Read Model object from an event specifically the EmployeeCreatedEvent. In this case, we have three events to handle: NET and Redis series, we: We need to set up our Dependency Injection system, our validation layer, and our Requests.

Chapter 3 : Real-World CQRS/ES with blog.quintoapp.com and Redis Part 3 - The Read Model

In Part 3 of our Real-World Angular Series, we've covered fetching data from the database with a Node API and manipulating and displaying the data in Angular. In the next part of the tutorial series, we'll tackle access management, displaying the admin events list, and developing an event details page with tabbed child components.

Published on July 17, Content series: This content is part of in the series: Real world Rails, Part 3 <https://> This content is part of the series: Real world Rails, Part 3 Stay tuned for additional content in this series. Ruby on Rails programming will spoil you. With Java frameworks, I would normally build a map between independent models and schemas. Frameworks like this are mapping frameworks. Frameworks that base the object model design on the structure of the database are called wrapping frameworks. But unlike most wrapping frameworks, Rails can discover the features of the object model by querying the database table. Instead of building complex queries, I can use the model to traverse relationships in Ruby instead of SQL. I get the simplicity of a wrapping framework with much of the power of a mapping framework. ActiveRecord is easy to use and easy to extend. Like any database framework, ActiveRecord lets me do many things that get me into trouble. I can fetch too many columns, or leave off important structural database features, like indexes or null constraints. You just need to know how to harden your application if you need it to scale. As you know, that command generates your model, migration, unit test, and even a default fixture. You should also consider your overall database design. Keep these things in mind: Rails will not insulate you from basic database performance issues. Your database needs information, often in the form of indexes, to perform well. Rails will not insulate you from data integrity problems. Though most Rails developers do not like to keep constraints in the database, you should consider things like nullable columns. Rails has convenient defaults for many elements. Sometimes, default attributes like the length of text fields are too large for most practical applications. Rails will not force you to create an efficient database design. Before you trudge ahead and dive deep into ActiveRecord, you should make sure that you have a strong foundation. Make sure that your index structure will work for you. You need not drop down into SQL to create an index -- you can simply use a migration. Often, we will wait until we measure a problem in a given query before we take action. This strategy keeps us from second-guessing the database engine. But in the case of users, we know that the table will quickly grow into the millions of users, and will be ineffective without an index on frequently searched columns. Two more common problems also relate to migrations. If you have strings and columns that should not be null, make sure you code your migration appropriately. Most DBAs database administrators might think that Rails has the wrong default for null columns: By default, columns can be null. If you want to create a column that cannot be null, you must explicitly add the parameter: And if you have a string column, make sure that you code an appropriate limit. By default, Rails migrations will encode a string column as a varchar You should do your best to maintain a database structure that reflects your application. Rather than have an unrestricted login, if your application limits logins to 10 characters, you should code your database appropriately, as in Listing 2: Coding migrations with limits and non-nullable columns t. With a little up-front work, you can save yourself a whole lot of time chasing data integrity problems later. While you are considering your database basics, also think about which pages are static, and thus easy to cache. Given a choice between optimizing a query and caching a page, caching the page will give you a far greater return, if you can stomach the complexity. Sometimes, pages or fragments are purely static, such as a list of states, or frequently asked questions. In those cases, caching is a slam dunk. Other times, you may decide to limit your complexity, and attack database performance instead. If you should decide to attack query performance, read on. That means the framework will wait to access a relationship until you actually access it. Take, for example, a member that has an address. You can open the console and type the command: But if you type member. Normally, this lazy design works well, because the persistence framework does not need to move as much data to load a member. But assume you wanted to access a list of members, and all of their addresses, as in Listing 5: Retrieving multiple members with addresses Member. Listing 6 tells the story: You get one query for all of the members, and another for each of the addresses. We retrieved three members, and you got four

queries. Most persistence frameworks solve this problem with eager associations. Rails is no exception. If you know that you will need to access a relationship, you can opt to include it with your initial query. If you changed the query to Member. You see one query that retrieves all of the members and addresses. With ActiveRecord, you can also nest the: For example, consider a Member that has many contacts, and a Contact that has one address. You can improve the performance a little by eagerly including: You can do better by including both, as in Listing 9: You can use the eager loading technique whenever you know that you will use relationships in a given query. That technique is the performance optimization technique that we use for ChangingThePresent. Generally, eager associations will be more than enough. Inheritance and Rails When most Rails developers encounter Rails for the first time, they are captivated. You simply create a type column on your database table and inherit any subclass from the parent. Rails will take care of the rest. For example, you might have a table called Customer that inherits from a class called Person. A customer has all of the columns of Person, plus a loyalty number and an order history. Listing 10 shows the beauty in the simplicity of the solution. The master table has all of the columns of the parent and all subclasses. The code is simple, and not repetitive. The queries are simple with good performance because you need not do any joins to access multiple subclasses, and ActiveRecord can use the type column to determine which records to return. In some ways, though, ActiveRecord inheritance is fairly limited. If you have an inheritance hierarchy that is too broad, inheritance will break down. For example, at ChangingThePresent, we have several types of content that each have names, short and long descriptions, some common presentation attributes, and several custom attributes. We would like causes, nonprofits, gifts, members, drives, registries, and many other types of objects to inherit from a common base class so we could treat all types of content in the same way. Exploring alternatives We experimented with three solutions to this problem. One, we would have each proper class in its own table, and use views to build a common table for content. We threw this solution out early, because Rails does not deal with database views very well. Our second solution was to use simple polymorphism. With this strategy, each proper subclass has its own table. We push common columns down into each table. For example, say I need a superclass called Content that has only a name property with Gift, Cause, and Nonprofit subclasses. Gift, Nonprofit, and Cause would all have a name property. Since Ruby is dynamically typed, they need not inherit from a common base class. They need only respond to the same set of methods. ChangingThePresent uses polymorphism in several places to provide common behavior, particularly when we are dealing with images. The third alternative is to provide a common capability, but using associations instead of inheritance. ActiveRecord has a feature called polymorphic associations that is ideal for attaching common behavior to a class without inheritance. You saw an example of a polymorphic association earlier in Address. I can use the same technique to attach my common attributes for content management instead of inheritance.

Chapter 4 : Watch The Real World Episodes on MTV | Season 13 () | TV Guide

Programming Java threads in the real world, Part 2 Design for thread safety In last month's column, I demonstrated a simple deadlock scenario using two nested synchronized blocks that acquired the same two locks, but in a different order.

List of public events starting in the future List of all public and private events admin access required Event details authentication required List of RSVPs associated with an event authentication required Open up the server api. Recall that we already created and required our Event and Rsvp mongoose schema in Part 2: We can now use those schema to execute MongoDB collection methods with mongoose. Add the following code to the API Routes section of the api. This endpoint does not require any authentication. We also want to pass a projection see first example in this doc. Projections state which fields we want returned in the documents that match our query. If no projection is specified, all fields are returned. We want an empty array if there are no events, since a lack of event documents simply means none have been created yet. We can implement this like so: We want the admin to be able to see and interact with a listing of public and private events. This authorized endpoint needs to have a parameter passed to it when called. The parameter should be the event ID so we can use the findById method. RSVPs in our app are transparent to all authenticated users; many people want to know if their friends are attending the same events they are. Fetching Events Now that we have API routes for fetching events, we need to access these routes in our Angular app so we can display events data. Open the service and add the following code: Unable to complete request. We also need AuthService to prompt login if no JWT is found when attempting to make an authenticated request. Finally, in order to declare the types for our event streams, we need the models EventModel and RsvpModel we created earlier. Basic Authentication in Part 2. Our HttpClient methods e. You can read more about this in the Angular Http Docs here. This does not require any mapping on our end as of Angular v4. A failed call checks the error message and prompts a fresh login if necessary, canceling the observable and producing an error if something else went wrong. We can now import the service in any of our components to use its methods. Alternatively, we could use route resolve to prevent routes from loading until the necessary API data has been returned, but this can give an app the appearance of sluggishness while navigating. Generate the loading component like so: You easily can make your own at loading. The template is very simple: We can style the host element using the special selector: Run the following command to generate the boilerplate: Each event has a start datetime and an end datetime. Start and end dates for a single event may be different days or the same day. We want a way to collapse same-day events into one date when displaying them in the UI. We also need to provide DatePipe in our app. The isLoading method uses an expression to check if the loading argument strictly evaluates to false. The eventDates method accepts start and end dates, then uses the date pipe to transform the dates into user-friendly strings. If the start and end dates are the same, only one date is returned. The eventDatesTimes method does something very similar, but with times as well. Lastly, the eventPast method accepts an eventEnd parameter and compares it to the current datetime, outputting a boolean that informs us if the event has already ended. Provide Date Pipe and Utility Service in App Module We can now import and provide the date pipe and our utility service in the app module. In AngularJS, we would have used built-in filters for this, such as filter and orderBy. Angular uses pipes to transform data, but no longer provides out-of-the-box pipes for filtering or sorting for reasons cited here. Due to the performance and minification impacts of this choice, we will not create custom pipes to implement filtering or sorting functionality. This would simply re-introduce the same problems the Angular team was attempting to solve by removing these filters. Instead, the appropriate approach is to use services. Run the following command to generate a service with the Angular CLI: We already provided DatePipe in our app. The search method accepts the array of objects to be filtered, a query to search for, any optional properties we want to exclude from searching either a single property string or an array of properties , and optionally, a date format string. The dateFormat should be one of the formats from the Angular DatePipe. This allows users to search for dates that are much less readable in the raw data. The developer can determine which format they want to be able to query. For

example, if UTC date strings or JavaScript Date objects are transformed, the user can query for Jan and receive results with a value that is actually T That way, users can see the way dates are displayed in your listing and this can inform the way they structure their query. This is done differently for various value types. The noSearchResults method simply accepts an array and a query and returns true if the array is empty and a query is present. Our orderByDate method accepts an array of objects, the property containing the date value we want to sort by, and an optional reverse argument to change the sort order from ascending to descending. If no property is passed, the array is returned unsorted. As written, this method expects an array of objects because this is how our data is structured in the RSVP app. An array of dates will not be sorted. You can easily update this method to support more robust array sorting if needed for other apps. We can then use the sort array method to re-order the array by date timestamp. We can now search as well as sort our event arrays by date in our components. Home Component Event List Our components should get and display lists of events. Now we need to subscribe to and display it in our components. Import FormsModule and add it to the imports array in the NgModule. Finally, we need a member to store our search query. We want to be able to reset the query with a button in the template, so the resetQuery method sets the query property to an empty string and resets the filteredEvents array back to the initial eventList acquired in the API call. Home Component Template Now open the home. If event data was successfully retrieved and there are events present in the event array, we want to show a search form and the event list, which can be filtered on the fly. The search input is two-way bound to the query using the [ngModel] directive. This means that changes to the query will be kept in sync whether they came from the UI or the class. We then have a button that can be used to resetQuery. The console should also log the error message. Our public events homepage should look something like this now: If there are no events available in the database, the homepage should show this message: When an error occurs fetching events data, the homepage should look like this: Without the API accessible, it will show an error. Auth0 can help you focus on what matters the most to you, the special features of your product. We offer a generous free tier so you can get started with modern authentication. Auth0 makes it easy to add authentication to your Angular application.

The GI Crew get on the showroom floor for E3 and realizes it isn't what it should be. Other funny moments occur as well Support Gi on Patreon - <https://Pat>.

As with many companies, team and personal objectives are attached to features. In the worst cases, features that ended up being half-completed or downright broken had to be reverted after the release, a task requiring surgical precision to juggle reverting patches, all introducing yet more complexity for team to deal with. Laforge knew there had to be a way to smooth out the currents in the troubled waters. He started by taking a closer look at the type of work the team was focusing on at various points in the release cycle. By conducting this exercise, he was able to identify key points in the schedule that affected the entire trainâ€™such as two beta releases that were largely ignored by customers, and could be eliminated. He crafted a schedule that not only reflected the reality of how the team worked, what they cared about, and how they interacted, but was also simpler and easier to communicate to the rest of the business. It may come as a surprise, but they accomplish this not with fancy Git feature branching and merge strategies. In fact, they use Subversion to this day: There was also investment in the technology required to support feature flagging; this allows features to be developed behind flags, which are then turned on or off as the feature progresses through various phases out to customers. Laforge also called out the importance of being able to easily revert features, which has a technical as well as a cultural component. The important thing for our team is we have the mentality that trunk is always shippable. In the end, Laforge says it comes down to cultivating shared values among the team. For Chrome, it was socializing and accepting that the ship-schedule needed to be the driving factor, and building infrastructure, culture, and the right type of process around that core value: Continuous Delivery Without Breaking the Entire Internet In contrast to makers of software that users interact with on their desktops, Dyn is a company which may not immediately ring a bell. But you have definitely used their products. Most of the teams had solved the deployment problem in their own way. Cheslock decided to standardize on a tool he was familiar with, because it allowed him to provide the benefits to the teams earlier, made building his team easier especially in the still-tight hiring climate, and made it easier to focus on the cultural changes that are more critical to a continuous delivery transformation: Of course, all of this is as automated as possible, including environment creation, compilation, and test job setup, so the pipeline is effectively self-hosting and self-configuring. If we have a bad day, then everybody on the internet is going to have a bad day. All of this starts when the developer commits and pushes a cookbook change and creates an associated pull-request. Status is reported back via the change requests, as well as through various other mechanisms. Cheslock says in addition to the safety requirements, they had to make the entire thing simple. Dyn still maintains a NOC staff, and developers must open a change request to deploy new infrastructure and the associated application into various environments. This is mainly due to the numerous options available for each deployment such as whether the deploy will affect a single machine, a type of machine in a data center, an entire data center, or entire regions. To help communicate what needs to be released, Dyn relies on a standardized versioning mechanism that the development and NOC staff understand, and they use pull-requests for performing the release, which aids in auditing. But the parallels between the journeys toward continuous delivery for these two organizations are staggeringly similar.

Chapter 6 : The Ultimate Real-world FCP FAQ, Part 3 : Apple Final Cut Pro Legacy Tutorial

When Chad meets his housemates on "The Mad Real World," he's a little surprised.

It is not recommended that you use your internal system drive to capture video to. Your internal drive contains the Mac operating system, and it is very busy constantly loading or referencing system resources. This causes you to drop frames while capturing or playing video files. For best results, you need either a secondary internal drive if using a Mac tower, or external drive, like a firewire or eSATA drive. The last version of the operating system that Apple recommends for FCP 4. People have had success in using the Mac OS To fix this you have two options. This way you ensure that you have a nice, proper working system. To get back to 7. And the only way to do that is an Archive and Install, or to erase and install. That is the beauty of this camera. So you will see the image, and the name, but the video file would have been deleted. If something is, then you know what the problem is. You can then just import the rest a few at a time, skipping that one. Try shortening them and see if that works. The solution is to re-import the footage with FCP 5. No worries, not all is lost. Much like the six million dollar man, we can rebuild him. If you are using a Mac then open up textedit. Create a new txt file. It will have 3 lines to it. The first line is the name of the last clip that was recorded. Back to the finder. Organize the folder by Date Last Modified. Find the name of the last file in this folder. If you have 16 clips in the folder it will be ag. Four numbers and two letters followed by. Take this name and place in the first line of your text file. On the next line of the text file type in "1. If you have 16 clips then the third line should read "17" again no quotes. Remember to add 1 to that number. Your txt file should look something like this: You should be good to go. The annoying thing is that all of the various. It does so without leaving all of these locked files on your computer. Credit for this solution goes to Randy Wedick. Drag the Final Cut Pro application to the Trash. In the Receipts folder, select the "FinalCutPro. Click the Date Modified column header so you can easily see all of the receipts that were installed at the same time as Final Cut Pro. Click the Name column header to sort the list alphabetically. Drag any other receipts whose names begin with "Final Cut Pro" to the Trash. When finished, use Software Update under the Apple menu to update your software to the latest version. Thanks to Apple for this one. Among the most frequent causes is having more than 64 QuickTime components installed. This is most likely due to installed third-party QuickTime Components. The components made by Apple will be marked with Apple copyright information. Drag any third-party non-Apple components to the desktop. Select the component in the Finder. In the Info window that appears, look at the Version info. The following components are definitely not third-party components; they are installed by Apple software, and should not be removed.

Chapter 7 : Real Housewives of Atlanta Season 10 Episode 21 (Reunion Part 3) – Mr. World Premiere

ActiveRecord is a fantastic persistence framework, but since the framework hides low-level details from you, it can be prone to performance problems. Discover the most common problems and how you can solve them.

Real-World Influences of Avatar Part 3: One of my favorite aspects of the show are all of the real-world people and events that influence the show. So I have decided, for your enjoyment, to discuss each nation in the show and its influences. Part 3 will be the Earth Kingdom. All of my sources are the Avatar wiki and Wikipedia, as well as the Avatar extras. Part 1 - Air Nomads: The Earth King during Avatar: Kuei ascended to throne at the age of four, but due to his youth, the kingdom was placed in the hands of Grand Secretariat of Ba Sing Se, Long Feng. Over the next 21 years, Long Feng manipulated Kuei and used him as a puppet ruler, effectively becoming the true ruler of the Kingdom. For a short time, Kuei was forced into exile after Azula took over the Earth Kingdom, but his rule was eventually restored. Kuei shares several similarities with Emperor Puyi, the last emperor of China. Like Kuei, Puyi rose to power at a very young age and he eventually became the ruler of the puppet state of Manchukuo, being used by others to retain power. He too remained trapped in the Forbidden City for many years. Hou-Ting was seen as a terrible ruler, levying very high taxes on the citizens and willing to execute anyone who spoke against her. She also reinstated the Dai Li agents, despite the fact that they helped Azula overthrow her father 70 years prior. Her actions lead to her eventual assassination in her own throne room as well as a peasant uprising within the Earth Kingdom. Cixi wore similar clothing to Hou-Ting, matching her hat and weird finger-claw-things. She is also seen as a despot by much of the world. The opinion is that she contributed to the eventual collapse and revolution of China. Whether she was assassinated or not is still unknown but some believe that it is a possibility. Another similarity can be found in how Cixi installed her nephew as the Guangxu Emperor, similar to how Hou-Ting was succeeded by her great-nephew Wu. Like Puyi, Wu was the last Earth King and was overthrown by another. In Legend of Korra, it is made clear that Wu will not be ruling the country, instead leaving that job to his advisors. On the day of his coronation, however, Kuvira announces her plans to become Emperor of the Earth Kingdom and he is exiled from his own nation. After Kuvira is defeated, Wu decides to abdicate and allow the Earth Kingdom to elect leaders for each independent state. This ends the line of the monarchy. The ruler that followed Hou-Ting and overthrew Wu was Kuvira. She refused to allow her nation to fall back into the hands of a foolish monarch and decided to form the Earth Empire, with her as its head. Kuvira, unlike other rulers of the Earth Kingdom, is not based off any traditional Chinese rulers, instead taking inspiration from events surrounding the World Wars. Kuvira set up a heavily fortified border with large walls that were constantly patrolled, similar to the Berlin Wall and she was known to send non-ethnic Earth Kingdom citizens to camps. Unlike Kuvira, however, Hitler did not build and pilot a massive mecha with the Dora attached. For me, the Dai Li were one of the creepiest parts of the show, and the draw inspiration from many places. This system has prisoners sent to forced labor camps. Some non-Chinese inspirations for the Dai Li may be the Roman Praetorian Guard and the Ottoman Janissaries, both forces intended to protect and serve their rulers, but who were known to control them if need be. Earthbending is based off of the Hung Gar style of Kung Fu. Toph emphasizes the importance of stances and staying rooted. Hung Gar also spends a significant amount of time teaching stances and staying rooted to the ground. Rock is a stubborn element and in order to bend it, you need to be like a rock yourself. Toph, however, is an exception in that her style of earthbending is based off of Southern Praying Mantis. The earthbending style featured in probending matches is based off of mixed martial arts fighting. It is based off of professional wrestling shows such as the WWE. I hope you all enjoyed this post.

Chapter 8 : Real-World Angular Series - Part 3: Fetching and Displaying API Data

Keep in mind that the Real World cast members are paid \$ a week. That may sound like plenty of money when you don't have to pay rent, but cast members likely have expenses back home if they don't want to return to find all their stuff in boxes on the curb, and rent was the only thing they didn't have to pay.

Map of Guatemala showing the shooting location and other important destinations. While many tourists come each year to visit the many temples, it never feels crowded. Make sure you have the time for a full hike around the park. If you come early in the morning the park opens at 6. Other animals like howler monkeys or coatimundi can be seen or heard as well. The park also holds a well-deserved reputation for exotic bird watching like toucans. There are various tours available like a sunrise tour or a canopy tour in which you soar along zip lines above the jungle. They found what they were looking for in Tikal, and spend five days of shooting at Temple IV, filming from the entranceway of the comb structure looking eastward back towards the Tikal Plaza. The Millennium Falcon is spotted on Yavin 4. Funny enough, the next shot has the Falcon flying from left to right, but in the same location. They only altered the angle slightly so that you can now see a Rebel guard scanning the ship next to a piece of wall from Temple IV. While the top of the Great Temple on Yavin 4 has not always been consistent within various sources, Temple IV, the temple that they filmed from, seemed to have been the inspiration for Al Williamson when he had to draw the Great Temple for Classic Star Wars 8. Entering the Great Temple – movie vs. The next shot of Yavin 4 that we see is that of the heroes standing in front of the entrance of the Great Temple. However, this was not filmed in Guatemala. The background is a matte painting and the live-action part was filmed in front of a hangar at a Royal Air Force base: Cardington Air Establishment in Bedfordshire, England. Setting up the lookout nest at the Rebel base. Yet due to the height and the peril of being on the border of the summit, it was scary to climb in. So when Lorne Peterson arrived to join the crew on the next day, they immediately coaxed him into the lookout nest, and he became the Rebel we see in these shots. The misty jungle shot that follows after the first time we see the lookout nest, was filmed early in the morning from Temple IV on the second day of filming. The stunning Atitlan Lake. What else to do in Guatemala? Besides Tikal it is well worth it to visit: The region is littered with Maya ruins, both discovered and undiscovered. Former capital city with a large legacy of colonial architecture. This lake is surrounded with three volcanoes and little Maya villages. This active volcano erupts on a regular basis, offering a very spectacular view of the lava and ash clouds. If you like the Maya ruins of Tikal, you will find more here. Map courtesy of Wikimedia Commons, adapted by Stefan Pfister. Recently he finished his Tourism Management study and he cannot wait to work in the tourism industry.

Here in Part 3, he covers QuickTime issues, timecode management, P2 tips and much more, including one of the most frequently questions of all: how to open new projects in older versions of FCP. Creative Cow Leader Shane Ross is one of a very small handful of the worlds most respected FCP experts.

Liberal Long Island cartoonist Judd meets with Arizona State University graduate Rachel , whom he immediately likes, but his enthusiasm is tempered upon learning that she is a Republican. At the house, they meet med student Pam and musician Mohammed. Bike messenger Puck shows up to the house later after having been arrested, and immediately makes an impression with his eccentricity. Pedro reveals his HIV status to the others by showing them his scrapbook as an educator. Although Pam and Cory react positively, Rachel is made uncomfortable by the revelation, and walks away, concerned over how living with someone with HIV will affect her. The cast attempts to address these issues during a house meeting, but Puck rejects their suggestions. Pedro and Rachel discuss her reaction to his scrapbook presentation in the previous episode, which he took as a sign of rejection. The housemates find communicating with Puck to be difficult. Puck goes to court to address his required drunk driving rehabilitation. He is later offended when only Rachel attends his soapbox derby , but the cast feel he is playing the victim for not being able to monopolize all of their time. Judd and Pam grow close, and spend time with Christopher, her long-distance boyfriend of eight years. Pedro, who is concerned about the gay jokes the latter two make over the phone, ponders his relationship with Sean. As Puck continues to resolve his legal difficulties, he finds a stray dog and takes it home. Cory sees a man named Geoff, but he may not be right for her. Pedro announces his engagement to Sean, and Puck dismisses it as a joke. Puck openly states that he dislikes Pedro, and feels no need to change for him or anyone, leading to an emotional group discussion among the others. After much soul-searching, the cast decides to evict Puck. Rachel discusses her rebellious relationship with her parents. Rachel and Pedro grow closer, and he accompanies her to Arizona to meet her family, and to speak at a local school. Puck meets with Cory, and complains about their eviction of him, prompting a discussion of his manipulation of others. The cast auditions three people to be their new housemate, and picks Jo Rhodes. An avid rock climber , Jo takes Rachel and Cory rock climbing with her after moving in. Cory and Steve accompany her to Lake Tahoe , where the restraining order is upheld. Pedro discusses his work and his future, while his health causes grave concern in the others. After he receives treatment for pneumonia , he improves, and the cast enjoys a horseback riding excursion. Rachel invites the cast to an Empower America conference, but when Judd comes away with objections to their politics, his comments offend her. Puck invites his former housemates to his soap box derby, but none are interested in attending. During the ceremony, which is held at the house, Pedro and Sean exchange vows and wedding bands. Meanwhile, Rachel and Jo continue to bond, and Mohammed concentrates on his music. As the housemates leave the house for good, they reflect on their time together, with some regretting that they did not spend more time together, but feel the experience has changed them for the better. Pedro Zamora fell ill and was hospitalized in New York in October He was eventually flown to Miami , and was diagnosed with progressive multifocal leukoencephalopathy , or PML. He received a phone call from President Bill Clinton , who thanked him for his work, and helped facilitate emigration of his older brothers, who were flown to Miami from Cuba, reuniting them for the first time in 14 years. The money from the benefit was used to form the Pedro Zamora Memorial Fund. He also chronicled his friendship with Zamora in his autobiographical graphic novel , Pedro and Me: Friendship, Loss and What I Learned. Judd and Pam , who fell in love during the vigil they kept over Pedro, married in , and as of , have two children. They eventually married, and as of May , have eight children. Los Angeles , and married on camera Betty, his fiancée and mother of his sons, Bogart and Rocco. In his spare time, he enjoys gardening and photography, though he prefers to model, whereas his partner works behind the camera. He was active in youth and mentoring organizations, and he and Kaplan served as foster parents to a 4-year-old girl.