

Lecture 3 - Before the Startup (Paul Graham)

YC Root Access · YC Root Access · 2014-10 · [source](#)

That short. Like, long introductions are no good. Sam knows. Alright, ready, everybody? I'm not gonna ask if the mic is working like in every talk so far. I'll just assume it's working. >> No! >> No, fuck! >> All right, well make it work somebody. >> It works, it works >> All right, this is like some kinda class tradition. All right, all right. I wrote out my talk and afterwards in a couple days I will like turn it from a talk into an essay and put it online. So, you don't have to take notes, just, just listen. All right. So, one of the advantages of having kids is that when you have to give advice to people, you can ask yourself, what would I tell my own kids? And actually, you find this really focuses you. So, even though my kids are little. My two year old today, when I asked what he was gonna be after two? He said a bat. The correct answer was three. But a bat is so much more interesting. So even though my kids are little, I already know what I would tell them about startups if they were in college. And so that is what I am gonna tell you. So, you're literally getting what I would give my own kids, since most of you are young enough to be my own kids. All right. So, startups are very counterintuitive. And I'm not sure exactly why, it could be simply because knowledge about them has not permeated our culture yet. But, whatever the reason, this is an area where you cannot trust you're intuitions all the time. It's like skiing in that way. Any of you guys learn to ski as adults? You know, when you're skiing, when you first try skiing and you want to slow down, your first impulse is to lean back, just like in everything else, but lean back on skis and you fly down the hill out of control, so, as I learned. So, part of learning to ski s learning to suppress that impulse. Eventually, you get new habits. But in the beginning there's this list of things you're trying to remember as you start down the hill. You know, like alternate feet, make s-turns, do not drag the inside foot, all this stuff. Well, startups are as unnatural as kiing and there is a similar list of stuff you have to remember for startups. And what I'm gonna give you today is the beginning of the list. The list of the counterintuitive stuff you have to remember to like prevent your existing instincts from leading you astray. The first thing on it, is the fact that I just mentioned that startups are so weird that if you follow your instincts they will lead you astray. If you remember nothing more than that. When you're about to make a mistake, you may at least pause before making it. When I was running Y Combinator, we used to joke that our function was to tell founders things they would ignore. And it's really true. Batch after batch, the YC partners warn founders about mistakes they are about to make. And the founders ignore them. And they come back a year later and say, I wish we'd listened. But, that dude is in their cap table and there's nothing they can do. Why do founders persistently ignore the partner's advice? Well, that's the thing about counterintuitive ideas, they contradict your intuitions, so they seem wrong. So, of course you're first impulse is to ignore them. And in fact that is not just he curse of Y Combinator, but to some extent our You don't need people to give you advice that doesn't surprise

you, right? If founder's existing intuitions gave them the right answers they wouldn't need us. That's why there are a lot of ski instructors and not a lot of running instructors. You don't see those two rds together, running instructor, as much as you see ski instructor. It's because skiing is counterintuitive. So sorta, what YC is, is like business ski nstructors, except for going up the slopes instead of down them. Well, ideally, you can however trust your instincts about people. You, your life so far hasn't been much like starting a startup, but all the interactions you've had with people are just like the interactions you've had with people in the business world. So, in fact one of the big mistakes that founders make is to not rust their intuitions about people enough. They meet someone who seems impressive, but about whom they feel some misgivings. And then later, when things blow up, they tell themselves, they say, you know, I knew there was something wrong about that guy, but I ignored it because he seemed so impressive. And there's a specific sub-case in business, especially if you come from an engineering background as I believe you all do. You think business is upposed to be this sorta slightly distasteful thing. And so when you meet people who seem smart, but somehow distasteful you think well, okay this must be normal for business. But it's not. Just like pick people the way you would pick people if you were picking friends. This is one of those rare cases where it works to be self indulgent. Work with people you genuinely like and respect and that you have known long enough to be sure. Because there's a lot of people who are really good at seeming likeable for a while. Just wait till your interests are opposed and then you'll see. All right. The second counterintuitive point is that, and this will, might come as a little bit of a disappointment. But what you need to succeed in a startup is not expertise in startups. That makes this class different from most other classes you take. You take a French class at he end of it you will earn how to speak French if you do the work. You may not sound exactly like a French person, but pretty close, right? This class can teach you about startups but hat is not what you need to know. What you need to know to suceed in a startup is not expertise in startups, what you need is expertise in your own users. Mark Zuckerberg did not succeed in Facebook because he was an expert in startups. He succeeded despite being a complete noob at startups. I mean, Facebook was first incorporated as a Florida LLC. Even you guys know better than that. He succeeded despite being a complete noob at startups because he understood his users very well. Most of you don't know the mechanics of raising an Angel round. Right, and if you feel bad about that, don't, because I can tell you Mark Zuckerberg probably doesn't know the mechanics of raising an Angel round either. If he was even paying attention when Ron Conway wrote him the big check, he has probably forgotten about it by now. In fact, I worry, it's not merely unnecessary for people to learn in detail about the mechanics of starting a startup, but possibly somewhat dangerous. Because another of the characteristic mistakes of young founders starting startups is to go through the motions of starting a startup. They come up with some plausible sounding idea. They raise funding at a nice valuation. They rent a nice office in Soma, hire a bunch of their friends. And then the next step, after rent a nice office in Soma and hire a bunch of their friends, is gradually realize how completely fucked they are. Because while imitating all the outward forms of starting a startup, they have neglected the one thing that's actually essential which is to make something people want. By the way, that's the only use of that swear word except for the initial one that was

involuntary. And I did check with Sam whether it would be okay. He said he had done it several times. I mean, used the word. >> We saw this happen so often, No, I mean people going through the motions of starting a startup. >> That we made up a name for it. Playing house. Eventually, I realized why it was happening. The reason young founders go through the motions of starting a startup is because that is what they have been trained to do for their whole lives up to this point. Think about what it takes to get into college. Extracurricular activities, check. Right? Even in college classes, most of the work you do is as artificial as running laps. And I am not attacking the educational system for being this way. Inevitably, the work that you do to learn something is gonna have some amount of fakiness to it. And if you measure people's performance, they will inevitably exploit the difference. To the degree that what you're measuring is largely an artifact of the fakiness. And I confess that I did this myself in college. In fact, here is a useful tip on getting good grades. I found that in a lot of classes there might only be 20 or 30 ideas that actually, had the right shape to make good exam questions. And so the way I studied for exams in these classes was not to master the material in the class. But to try and figure out what the exam questions would be and work out the answers in advance. Unlike, for me the test was not like what my answers would be on the exam. For me the test was. Which of my exam questions would turn up on the exam, right? So I would get my grade instantly. I would walk into the exam and look at the questions and see how many I got right essentially. >> It works in a lot of classes, especially CS classes. I remember Automata Theory, there's only a few things that make sense to ask about Automata Theory. So, it's not surprising that after being effectively trained for their whole lives to play such games, young founders first impulse upon starting a startup is to figure out what the tricks are. For this new game, what are the extracurricular activities of startups, what are the things that I have to do. They always wanna know since, since apparently the measure of success for a startup is fundraisingn another noob mistake. They always wanna know, what are the tricks for convincing investors. And we have to tell them, the best way to convince investors, is to start a startup that's actually doing well, meaning rowing fast, and then simply tell investors so. So then they ask, okay, what are the tricks for growing fast? Alright? And this is exacerbated by the existence of this term, growth hacks. Right? Whenever you hear anybody talk about growth hacks, just mentally translate it in your mind to bullshit. Because what we tell them is the way to make your startup grow is to make something that users really love. Right, and then tell them about it. So that's what you have to do. That is growth hacks right here. So many of the conversations the YC partners have with the founders begin with the founders saying a sentence that begins with, how do I. And the partner is answering with a sentence that begins with, just. Right, why do they make things so complicated? The reason, I realized, after years of being puzzled by this, is they're looking for the trick, they've been trained to look for the trick. So, this is the third thing. The third counterintuitive thing to remember about startups. Starting a startup is where gaming the system stops working. Gaming the system may continue to work if you go to work for a big company. Depending on how broken the company is you may be able to succeed by sucking up to the right person. Giving the impression of productivity by sending emails late at night. Or if your smart enough, changing the clock on your computer, cause whose going to check the headers,

right? But, I like an audience I tell jokes like that to and they laugh. Over in the business school, headers, okay. God this thing is being recorded, I just realized that. All right, from now on, we are sticking strictly to the script. & All right. But in startups, that does not work. There is no boss to trick. There's no, how can you trick people if there's no one to trick? There's only users. And all users care about is whether your software. Does what they want, right? They're like sharks. Sharks are, like, too stupid to fool. You can't, like, wave a red flag at a shark and fool it, it's like, meat or no meat. So, you have to have something people want. And you only prosper to the extent that you do. The dangerous thing is, particularly for you guys, the dangerous thing is that faking does work to some extent with investors. You, if you're really good at knowing what you're talking about. You can fool investors for one, maybe two rounds of funding. But it's not in your interest to. I mean, you're all doing this for equity. You're playing a confidence trick on yourself. You're wasting your own time, because the startup is doomed. And all you're doing is, you're just gonna waste your time writing it down. So, stop looking for the trick. There are tricks in the startups, as there are in any domain. But they are an order of magnitude less important than solving the real problem. Someone who knows zero about fund raising but has made something users really love will have an easier time raising money. Than someone who knows every trick in the book, but has a flat usage graph. Though in a sense it's bad news that gaming the startups stops working now. In a sense that you're deprived of one of your most powerful weapons and after all something you have spent 20 years mastering. It is something, I find it very exciting. That there even exists parts of the world where gaming the system is not how you win. I would have been really excited in college if I had explicitly realized that here were parts of the world where gaming the system matters less than others. And some where it hardly matters at all. But there are, and this is one of the most important things to think about when you're planning your future. Got that, okay. How do you win at each type of the work? And what do you want to win by doing? So, that brings us to our fourth counterintuitive point. Startups are all consuming. If you start a start up, it will take over your life to a degree that you cannot imagine. And if it succeeds, it will take over your life for a long time. For several years, at the very least. Maybe a decade. Maybe the rest of your working life. So there's a real opportunity cost here, it may seem to you that Larry Page has an enviable life, but here are parts of it that are definitely unenviable. The way the world looks to him, is that he started running as fast as he could at age 25. And he has not stopped to catch his breath since. Every day, shit happens within the Google empire that only the emperor can deal with. And he, as the emperor, has to deal with it. If he goes on vacation for even a week, a whole backlog of shit accumulates. And he has to bear this uncomplainingly, because, number one, as the company's daddy, he can never show fear or weakness. And number two, if you're a billionaire, you get zero, actually less than zero sympathy if you complain about having a difficult life. Which has this strange side effect that the difficulty of being a successful startup founder is concealed from almost everyone who's done it. People who win the hundred meters in the Olympics, like they walk up to them and they're going like, right. Larry Page is doing that too, but you never get to see it. All right, where are we? Y Combinator has now funded several companies that could be called big successes. And in every single case, the founders say the same

thing, it never gets any easier. The nature of the problems change. So, you may be worrying about more glamorous problems like construction delays in your new London offices, rather than the broken air conditioner in your studio apartment. But the total volume of worry never decreases, if anything, it increases. Starting a successful startup is similar to having kids, in that it's like a button you press, that changes your life irrevocably. And while it is, like, honestly the best thing in the world having kids. If you take away one thing from this lecture, remember that. && There are a lot of things that are easier to do after you, before you have kids, then after. Many of which will make you a better parent, when you do have kids. And so, in rich countries, most people delay pushing the button for a while. And I'm sure you are all intimately familiar with that procedure. Yet, when it comes to starting startups, a lot of people seem to think they're supposed to start them in college. Are you crazy? And what are the universities thinking? They go out of their way to ensure that their students are well supplied with contraceptives. And yet they're starting up entrepreneurship programs and startup incubators left and right. && To be fair, the universities have their hand forced here. A lot of incoming students are interested in startups. Universities are at least defacto, supposed to prepare you for your career. And so, if you're interested in startups, it seems like universities are supposed to teach you about startups. And if they don't, maybe they loose applicants to universities. You do claim to do that. So, can universities teach you about startups? Well, if not what are we doing here? && Yes and no, as I explain, they can teach you about startups, but this is not what you need to know. Essentially, what universe, if you wanna learn French, universities can teach you linguistics. That's what this is. This is a linguistics class. Right? We're teaching you about how to learn languages. And what you need to know is like, how to learn a particular language? What you need to know, are the needs of your own users. You can't learn those, until you actually start the company. And starting a company, which means that starting a startup is something, and you can intrinsically only learn by doing it. And you can't do that in college for the reason I just explained, that startups take over your entire life. If you start a startup in college, If you start up a startup as a student, you can't start up a startup as a student, because if you start a startup you're not a student anymore. You may be nominally a student, but you won't even be that for much longer. So, given this dichotomy, which of the two paths hould you take? Be a real student and not start a startup or start a real startup and not be a student. Well I can answer that one for you. I'm talking to my own kids here. Do not start a startup in college. I hope I'm not disappointing anyone, seriously, honestly. Starting a startup could be a component of a good life for a lot of ambitious people, but this is just part of a much bigger problem that your trying to solve. How to have a good life, right, and though starting a startup could be a good thing to do at some point, 20 is not he optimal time to do it. There are things, that, that you can't, there are things you can do in your early 20's, that you cannot do as well before or after, like plunge deeply into projects on a whim that seem like they have no payoff. Or travel super cheaply with no sense of a deadline. In fact those are really just isomorphic shapes in different domains. For unambitious people this ort of thing can be the dreaded failure to launch. But for the ambitious ones, it's a really valuable sort of exploration. And if you start a startup at 20, and you're sufficiently successful, you will never get to do it. Mark Zuckerberg will

never get to bum around a foreign country. He can, if he goes to a foreign country, it's either as a defacto state visit, or, like, he's hiding out incognito at he George Cinc in Paris. Right? But he's never gonna get to just, like, backpack around Thailand if that's still what people do. Do people still backpack around Thailand? >> Yes. >> Okay, well there, that's the first real sign of enthusiasm I've seen from this audience. I should have given this talk in Thailand. >> All right, he can do things that you can't do, like charter jets to fly him to foreign countries, really big jets. But success has taken a lot of the serendipity out of his life. Facebook is running him as much as he is running Facebook. And while it can be really cool to be in the grip of some project you consider your life's work, there are advantages to serendipity. And among other things, it gives you more options to choose your life's work from. There's not even a trade-off here. You're not sacrificing anything if you forego starting a startup at 20, cuz you'll be more likely to succeed if you wait. In the unlikely case that you're 20, like astronomically unlikely case, that you're 20 and you have some side project that akes off like Facebook did. Then you face the choice of either running with it or not, and maybe it's reasonable to run with it. But usually the way startups take off is for the founders to make them take off. It's gratuitously stupid to do that at 20. So should you do it at any age? I realize, I've made Starting a startup sound kind of hard. If I haven't, let me try again. Starting a startup is really hard. What if it's too hard? What if you're not up to this challenge? The answer is the fifth counter-intuitive point. You can't tell. Your life, so far, has given you some idea what your prospects might be, if you wanted to become a mathematician, or a professional football player. Boy, it's not every audience you could say that to. But unless you have had a very strange life, indeed, you have not done much that's like starting a startup. Meaning, starting a startup will change you a lot if it works out. So, what you're trying to estimate is not just what you are, but what you could become, and who can do that? Well, not me, for the last nine years, it was my job to try and guess whether people would. Guess is, I wrote predicted here and it came out as guess. That's a very informative Freudian slip. >> Seriously, it's easy to tell how smart people are in ten minutes. You know, hit a few tennis balls over the net and o they hit them back at you or into the net. But, the hard part was, and the most important part was predicting how tough and ambitious they would become. There may be no one at this point who has more experience than me at doing this. And I can tell you how much an expert can know about that. The answer is, not much. I learned from experience to keep a completely open mind about which startup in each batch, which startups would turn out to be the stars. The founders sometimes thought they knew. Some arrived feeling confident that they would ace Y Combinator, just as they had aced every one of the few, easy, artificial tests they had faced in life so far. Others arrived wondering what mistake could cause them to be admitted and hoping that no one would iscover it. But there's little to no correlation between these attitudes in how things turn out. I've read the same is true in the military that he swaggering recruits are no more likely to turn out to be really tough than the quiet ones. And probably for the same reason that the tests are so different from the tests in people's previous lives. If you're absolutely terrified of starting a startup, you probably shouldn't do it, unless you're one of those people who gets off on doing things you're afraid of. Otherwise, if you're merely unsure of whether you're gonna be able to do it, the only way to find

out is to try, just not now. So, if you wanna startup one day, if you wanna start up one day, what do you do now in college? There are only two things you need initially, an idea, and co-founders. And the MO for getting them both is the same, which leads to our sixth, sixth and last counter-intuitive point. The way to get startup ideas, is not to try to think of startup ideas. I've written a whole essay on this and I'm not gonna repeat the whole thing here. But the short version is that if you make a conscious effort to try and think of startup ideas, you will think of ideas that are not only bad, but bad and plausible sounding. Meaning you and everybody else will be fooled by them, and you'll waste a lot of time before realizing they're no good. The way to come up with good startup ideas is to take a step back. Instead of trying to make a conscious effort to think of startup ideas, turn your brain into the type that has startup ideas unconsciously. In fact, so unconsciously that you don't even realize at first that hey're startup ideas. This is not only possible, Yahoo, Google, Facebook, and Apple all got started this way. None of these companies were ven supposed to be companies at first, they were all just side projects. The very best ideas almost have to start as side projects, because they're always such outliers that your conscious mind would reject. As ideas for companies. Okay? So, how do you turn your mind into the kind that has startup ideas unconsciously? 1, learn about things that matter. 2, work on problems that interest you. 3, with people you like and respect. That third part incidentally is how you get co-founders at he same time as the idea. The first time I wrote that paragraph, instead of learn a lot about hings that matter, I wrote become good at some technology. But that prescription, though sufficient, is too narrow. What was special about Brian Chesky and Joe Gebbia from Airbnb was not they were xperts in technology. They went to art school. They were experts in design, and perhaps even more importantly, they were really good at organizing people and getting projects done. So, you don't have to work on technology per se, so long as you work on things that stretch you. What kind of things are those? Now, that is very hard to answer in the general case. History is full of examples of young people who were working on problems that no one else, at the time, thought were important. And in particular that their parents didn't think they were important. On the other hand, history is even fuller of examples of parents who thought their kids were wasting their time, and who were right, so. >> How do you know if you're working on real stuff? I mean when Twitch TV switched from being Justin Dot TV to Twitch TV, and they were gonna broadcast people playing video games, I was like what? But, turned out to be a good business. Well, I know, I know how I know, real problems are interesting. And I'm self-indulgent. I like, I'm always interested in working on interesting things even if no one else cares about hem. And I find it very hard to make myself work on boring things even if they're supposed to be important. My life is full of case after case where I worked on things just cuz I was interested. And they turned out to be useful later in some worldly way. Y Combinator itself is omething I only did because it seemed interesting. So, I seem to have some sort of internal compass that helps me out. But, this is, for you, not me and I don't know what you have in your heads. Maybe if I think more about it I could come up with some heuristics for recognizing enuinely interesting ideas. But for now, all I can give you is the hopelessly question begging advice, incidentally, this is the actual meaning of the phrase, begging the question. The hopelessly question begging advice that if you're interested in

genuinely interesting problems, gratifying your interest energetically is the best way to prepare yourself for a start-up. And, for that matter, probably the best way to live. But although I can't explain the general case what counts as an interesting problem, I can tell you about a large subset of them. If you think of technology as something that's spreading like a sort of fractal stain, every point on the edge that's moving, represents an interesting problem. Steam engines is not so much, although maybe, you never know. So, one guaranteed way to turn your mind into the type that startup ideas form in unconsciously is to get yourself to the leading edge of some technology, to, as Paul Buchheit put it, live in the future. And when you get there, ideas that seem uncannily prescient to some people will seem obvious to you. You may not realize they're startup ideas, but you'll know they're something that ought to exist. For example, back at Harvard, back in the mid 90s, a fellow grad student of my friends Robert and Trevor, wrote his own voiceover IP software. It wasn't meant to be a startup. He never tried to turn it into one. He just wanted to talk to his girlfriend in Taiwan without paying for long distance calls. And since he was an expert on networks, it seemed obvious to him that the thing to do was to turn the sound into packets and ship them over the Internet for free. Why didn't everybody do this? Well because they weren't good at writing that kind of software. He never did anything with it. He never tried to turn it into a startup. But that is how all the best startups tend to happen. So, strangely enough, the optimal thing to do in college if you wanna be a successful startup founder, is not some sort of new vocational version of college focused on entrepreneurship. It's the classic version of college as education for its own sake. If you wanna start a startup, what you should do in college is learn powerful things. And if you have genuine intellectual curiosity, that's what you'll naturally tend to do if you just follow your own inclinations. The component of entrepreneurship, I can never quite say that word with a straight face, that really matters is domain expertise. Larry Page is Larry Page cuz he was an expert on search. And the way he became an expert on search, was because he was genuinely interested in it, not because of some ulterior motive. At its best, starting a startup is merely an ulterior motive for curiosity and you'll do it best if you introduce the ulterior motive at the end of the process. So here, is the ultimate advice for young would-be startup founders reduced to two words, just learn. All right, how much time do we have left? && 18 minutes. && 18 minutes of questions, good god, do you guys have the questions? && Sure, we'll start with, two questions online or we can start with others. && You guys are in charge, whatever you wanna do. && Okay. So, to start with, online questions. The most voted question today was- && Do I have to repeat them, by the way? && Yeah. && Okay. && How can a non-technical founder most effectively contribute to a startup? && How can a non-technical founder most effectively contribute to a startup? Well, if the startup is, if the startup is working in some domain. If it's not a pure technology startup, but is working in some very specific domain, like, if it's Uber, right? And the, and the non-technical founder was an expert in the limo business, then, actually, the non-technical founder would probably be doing most of the work, recruiting drivers and doing whatever else Uber has to do, right? And the technical founder will be just writing the iPhone app, which is probably less, well, iPhone and Android app, hm, which is less than half of it. If it's a purely technology startup, the, the non-technical founder does sales. And brings

coffee and cheeseburgers to the programmer. >> Okay, next. >> Oh, audience? Okay, audience, yes? >> Do you see, any value in business school for people who want to pursue entrepreneur, entrepreneurship? >> Do I see any value in. >> Or, if so, what value, I guess? I'm hoping that's not. >> Well, we probably won't have to get to the second question. >> Which I suppose is the answer to your first question. Okay, so the question is, is there any value in business school if you're interested in entrepreneurship? And, if so, what? So, basically, no. I mean, it sounds undiplomatic, but the point, what business school was designed for is to teach people management, right? And management is a problem that you only have in a startup if you're sufficiently successful. So, really, what you need to, what you need to know early on to make a startup successful, is developing products. You'd be better off going to design school, if you want to go to some sort of school. Although frankly, the way to learn how to do it is just to do it, you know? One of the things I got wrong early on, is that I advised people who were interested in starting a startup to go work for some other company for a few years before starting their own. But, honestly, the best way to learn how to start a startup is to try and start it. You may not be successful, but you'll learn faster if you just do it. So, not really. Business schools are trying really hard to do this, but honestly, like, they were, they were designed to train the officer corps of large companies, right? Which is what business seemed to be, back when it was a choice of either the officer corps of large companies or Joe's Shoe Store. And then, then there was this new thing, Apple, which started out as small as Joe's Shoe Store and then, like, turns into this giant mega-company. But they weren't designed for that world. And, like, they're good at what they're good at, they should just be, do that and screw this whole entrepreneurship thing, just cuz it's cool. Yes? >> You said, management is a problem that you have only when you're successful. >> Yes. >> So what about those first two or three people as a founder that you're trying to manage? >> Okay, so management is a problem only if you're successful. What about those first two or three people? Well, ideally, you're successful before you even hire two or three people, right? Didn't you say, Sam, that Airbnb took five months to hire their first employee? So ideally, you don't have even two or three people for quite a while. When you do, you can, you can sort of, the, the first hires in a startup are almost like founders. They should be motivated by the same things. They can't be people, they can't be people you have to, like, manage, right? This is like that, this is not, like, The Office or something like that. These are your, these have to be your peers, really. You shouldn't have to manage them much. >> So it's just a big no-no, like if someone has to be managed, no way they can be on the founding team? >> Well, never say never. I should repeat all these questions, so if someone has to manage, no way they should be on the founding team? In, in the case where you're doing something where you need some sort of super advanced technical thing. And there's some boffin, if you know that word, who knows this thing and nothing else in the world including, like how to wipe his mouth. Then it may be to your advantage to hire said boffin and wipe his mouth for him. But. >> Okay. >> As a general rule, you want people who are, sort of self-motivated early on. They should be just like founders. Yes, questions? In the far back? >> Do you think we're currently in a bubble? >> Do I think we're currently in a bubble? Okay, so I'll give you two answers to this question.

One, ask me questions that are useful to this audience, cuz these people are here to learn how to start startups and I have in my head, like, more data about that than ybody else. And you're asking me the kind of questions a reporter does cuz they can't think of anything interesting to ask. But I will answer your question. There is a difference between prices merely being high and a bubble. A bubble is a very specific form of prices being high, where people knowingly pay high prices for something in the hope that they will be able to unload it later on some greater fool. Right? And that's what happened in the late 90s. Like, VCs knowingly invested in bullshit startups, thinking that they would be able to take those things public, and unload them onto the retail investors before everything blew up. And I was there, I was there for that, at the epicenter of it all. And that is not what's happening today. Prices are high, valuations are high. But valuations being high does not mean a bubble. Every, every commodity has prices that go up and down in some sort of sine wave. Definitely, prices are high. And so we tell people, if you raise money, don't think the next time you raise money it's gonna be so easy. Who knows, maybe between now and then the Chinese economy will have exploded and there's another giant disastrous recession. Who knows, assume the worst, but bubble, no. Yes? && I'm noticing a trend among young people and successful entrepreneurs, where they don't want to start one more great company, they want to start like 20. And so you're starting to see the rise of these labs attempts where they're gonna try to launch a bunch of stuff. I don't know a really stellar example yet but. && Like ideal? && No like, like Idea lab, like new one. There's && Oh, yeah yeah. So there's this new thing where people start labs that are supposed to spin off startups, it might work. That's how Twitter started. In fact I meant Ideo Lab, not Idea, that was another Freudian slip. Because Twitter was not Twitter at first. Twitter was the side project. Side project at a company called Ideo that was supposed to be in the podcasting business. You're like podcasting business, do those words even grammatically fit together? The answer turned out to be no, as Evan discovered. But like as a side project they spun off Twitter and boy was that a dog wagging tail. So people are starting these things that are supposed to spin off startups, will it work? Quite possibly, quite possibly, if the right people do it. You can't do it yet though, cuz you have to do it with your own money. Yes, far back? && Thanks so much. I hate to step into sort of a gendered pitfall, but what advice do you have for female cofounders as they pursuing funding? && Female cofounders when you're pursuing funding, well it probably is true that women have a harder time raising money, right? I've noticed this in empirically, and, Jessica is just about to publish a bunch of interviews with female founders. And a lot of them said that they thought they had a hard time, a harder time raising money too. So, what I would, you remember I said, like the way to raise money? Make your startup actually do well. Well, that's just especially true in that case. If have any, if you, if you miss the ideal target from the VC's point of view in any respect, the way to solve that problem is make the startup do really well. So in fact, there was a point like a year or two ago when I tweeted this growth graph of this company and I didn't say who they were. But I knew it would start people asking and it was actually a female founded startup that was having trouble raising money but their growth graphs were stupendous. And so I tweeted it, knowing that, like, all these VC's would start asking me who is that, right? And, like, growth graphs have no gender, right? So if they see the

growth graph first, let them fall in love with it. Right? So do well, which is good advice for startups generally. Yes? && What would you learn in college right now? && What? && What would you learn in college right now? && What would I learn in college? Hm, literary theory? No, just kidding. && Let's see, well, you know, honestly, I think I might try and study physics. I feel that's the thing I feel like I missed. For some reason, like, I was all excited, when I was a kid, computers were the thing, maybe they still are, right? So I sort of got very excited about writing code. And you could do things with, you could do, you could write real programs in your bedroom. You can't like, build real linear accelerators. Well maybe you can, so maybe, physics. I feel kind of, I always, I have this, I sort of look long, I notice I sort of look longingly at physics. So maybe. But I don't know if that's, what am I saying? I'm saying, I was about to say, I don't know if that's going to be helpful starting a startup and I've just told you to follow your own curiosity! So who cares if it's helpful? It'll turn out to be helpful. Questions? && Another question online, what are your recurring systems in your work and personal life that make you efficient? Oh, boy, Okay. So what are my recurring systems in my work and personal life that make me efficient? Well, having kids is a good way to be efficient. Cuz you have no time left, so if you wanna get anything done, the amount of done you do per time is high. && So, actually many parents, many startup founders who have kids have made that point explicitly. It causes you to focus, because you have no choice. Let's see but that's not, I wouldn't actually recommend you having kids, just to have you more focused so, let's see, you know, I don't think I'm very efficient. I have two ways of getting work done. One is, like, during Y Combinator, the way I worked on Y Combinator was, I was forced to, right? Like, there was just, I had to set the application deadline. And then people would apply, and then there would be all of these applications that I'd have to respond to by a certain time, so I had to read them. And I knew if I read them badly we would get bad startups, so I tried really hard to read them well, right? So I set up this situation that forced me to work. The other kind of work I do is like writing essays. And I do that involuntarily. I'm like walking down the street and the essay starts writing itself in my head. And so really I either force myself to work on less exciting things or, I can't help working on exciting things, and I don't have any useful techniques for making myself efficient. Sorry. If you work on things you like you don't have to force yourself to be efficient. Yes? && When do you think is a good time to turn a side project into a startup? && When is a good time to turn a side project into a startup? You will know. Right? && What do you have the properties && It starts, you, you when you okay, the question is when you turn a startup into, a side project into a real startup. You will know that it's becoming a real startup when it takes over an alarming percentage of your life, right? Like when you find like, my God, I've just spent all day working on this thing that's supposed to be a side project, I'm gonna fail all my classes. What am I gonna do? Right? Then maybe it's turning into a startup. Yes? Wait, I already answered your question. I should ask, I should let somebody else ask one. I may get back to you. Yes? && I, I know you talked a lot earlier about you'll know when you start doing extremely well. But I feel like in a lot of cases, it's just you have a grey line where it's like, hey you have some users but maybe not as explosive growth chart as to the right. What would you do? Or what would you recommend doing in those situations specifically allocating time

and resources, how do you balance in? >> Okay, when a startup is out of growing but not much. >> Mm-hm. >> Didn't you tell them they were supposed to redo things that don't scale? >> Yeah. >> You sir, have not done the readings. >> And you are busted therefore. Because I wrote a whole essay in answer to that question. That is it. Do things that don't scale. Just go read that. Cuz I can't remember everything I said. >> But it's about exactly that problem. >> Yes, back there. >> What kind of startup should not go through incubation, in your opinion? >> What kinda, you mean. >> Well, yeah. >> Do you mean incubation YC? >> No, yeah, YC or. >> What kind of startup should not through YC. Definitely any that will fail. >> And, or, if you'll succeed but you're an intolerable person that also, Sam would probably sooner do without. Short of that, I cannot think of any, because most, a large percentage of, founders are often surprised by how large a percentage of the problems that startups have are the same regardless of what kind of thing they're working on. And those tend to be kind of problems that YC helps with the most not the ones that are domain specific. So I can't, is there, can you think of, class of startups that YC wouldn't work for? I can't. I mean like we had fission and fusion startups in the last batch. Rather they had, yes? >> You mentioned that it's good advice to learn a lot about something that matters? >> Yes. >> Some good strategies to figure out what matters? >> Well, If you think of, okay, so how do you figure out what matters? If you think of technology as something that's spreading as a sort of racial stain, anything on the edge represents an interesting idea. It sounds familiar. So how do you figure out hmm? Like I said, that was the problem. You have correctly identified, the thing where I didn't really answer the question, where I gave this question begging answer. I said, like I, just like, I'm interested in interesting things. If you're interested in interesting things, work on them and things will all work out, right? But how do you tell what is a real problem? I don't know. That's like important enough to write a whole essay about, and I don't know the answer. I probably should write something about that. But I don't know! I don't know. I figured out a technique for detecting whether you have a taste for genuinely interesting problems, which is whether you find working on boring things intolerable, right? And there are known boring things, like literary theory and working in middle management in some large company. So if you could tolerate those things, then you must not, you must either have stupendous self discipline, or you don't have a taste for genuinely interesting problems, and vice versa. Yes? >> One more question. >> Okay, one more question, it better be good. >> I was wondering out, you know like Snapchat, like, makes money >> Snapchat? What do I know about Snapchat? >> We didn't fund them. How about another question? All right go ahead. >> You talked about hiring employees you like. But it seems like that could read into a monoculture, and there are a lot of benefits in being a monoculture, but how do you feel about the blind spots that arise? >> Okay, so if you hire people who you like you might get a monoculture, and how do you deal with the blind spots that arise? Starting a startup is something where many things will be going wrong. You can't expect it to be perfect, and the advantages of hiring people that you know and like are far greater than, you know, the small disadvantages of

having some monoculture. You look at it empirically all the most successful startups someone just like hires all their pals out of college. All right you guys, thank you.