

Build Your Own Eval Tools With Notebooks!

73 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=AQKUWPKBk00](https://www.youtube.com/watch?v=AQKUWPKBk00)

<https://www.youtube.com/watch?v=aqKUwPKBk00>

SUMMARY

The discussion revolves around the evolution of data science practices, particularly focusing on the importance of thoroughly examining data rather than merely visualizing it. The speaker emphasizes the need for critical analysis in data science, illustrated through the analysis of the ChickWeight dataset, and introduces MIMO, a tool that enhances data exploration and interactivity in notebooks.

- Importance of critically examining data rather than just visualizing it.*
- The ChickWeight dataset serves as a teaching example to illustrate data analysis pitfalls.*
- MIMO provides interactive charts that allow users to explore data more deeply.*
- The platform supports reactive programming, automatically updating dependent cells when changes occur.*
- MIMO's architecture allows for flexible cell organization and dependency management.*
- The introduction of widgets and custom UI elements enhances user interaction with data.*
- MIMO supports various data sources and allows for integration with AI tools for enhanced functionality.*
- The speaker encourages creativity and exploration in data analysis, highlighting the potential of MIMO to facilitate innovative workflows.*

Um I started in data science like I don't know 10 15 years ago maybe it's been a while and nowadays when people talk about eval people say like look at your data look at your data look at your data it's a good like lesson people should definitely do it but the thing that always struck me that's interesting about it is that I had been doing that for like ages 15 years ago it was also a problem and the best way to explain the problem in my mind is to also talk about this one data set that I I tended to use for teaching this is the chickweight data set for those of you So remember this came with the R language. It's like one of those data sets that R came packaged with. And the data set has weight, it has time, it has a specific chicken. So different chickens are eating a specific diet over time so to say, right? So this chicken had this diet and as time moved forward, this chicken started gaining some weight.

Like that's the basically the story. And there's lots and lots of these chickens, right? So as I scroll over here, you can see lots of these chickens. Basically, um, and note, by the way, what we're looking at here is a MIMO notebook. Whenever you output a data frame, we get a nice interactive widget with lots of bells and whistles. There's a button over here. We just added it. So, what I could do is I could explore, let's say, the diet column. There's all sorts of descriptive statistics. There's like pretty charts and stuff, especially if you've got a time series. This is very nice. But, of course, we're not going to like stare at a data frame like this.

What you typically would do is you would make a chart. So, if you wanted to maybe summarize the data a bit, you would do something that looks a little bit like this. You would basically say like, look, for every chicken over

time, we've got this little dot. So, we've got time on the x-axis over here. We've got weight on the y-axis. And then for all of these different diets, we can also make like an average line. So, we can see that there's this diet 3 over here that seems to be doing quite well compared to some of these other diets.

And this is just doing summary statistics. You could also apply TensorFlow or scikitlearn if you want to do something fancy and predictive. But what a lot of people would do in data science is they would sort of look at this and go like okay diet 3 is best uh business case closed. And we have a chart here that shows us that that's accurate. And I would argue that you know this is a common practice but what I've just done is not looking at your data. What I've just done is staring at your data.

Because the number one thing that people always forgot to do whenever they were looking at a chart is that you shouldn't just look at it and then move on. What you should do is you should actually force yourself to look at the damn thing for like 5 minutes because nine times out of 10 something weird is happening but you have to look for it. And in this case because we're using Altera we can actually do a cool thing.

We can add interactive on the chart object itself. And when we do that uh we actually get a chart that we can move around and we get a chart that we can zoom into. And when you do that, suddenly something interesting happens because it seems like there are some chickens over here that are actually losing weight. And you know the presumably this is like a data set with chickens from like a feed lot. So that shouldn't really be happening. And if you squint your eyes, you can actually see another instance of that here. There's a I don't know exact.

You can't see right now if this chicken eventually lost weight or if it was a different one, but you do see that there's a point over here that's lower than anything that happened before. So, there are also chickens that are losing weight. And again, um that's just kind of weird. That should maybe not happen like that. That's kind of a smell there, right? And whenever you smell something, you should chase it. But one lesson here is you can only chase a smell if you take the time to sniff.

And if you took this effort, like if you noticed this one phenomenon, you could make a different chart uh which is this one. And then you could outline all these different chickens and then you would definitely notice that some chickens actually died prematurely. So there's one chicken over here that died prematurely. There's another one here, another one here. And like five chickens out of this entire group actually died before let's say uh the final date. Given that, what does this line actually represent?

I mean, yeah, you're going to calculate the mean of all the chickens over here, but you're going to not count the five chickens that died prematurely. Therefore, this line doesn't matter. And the only way you've got, and I I want to stress this, the only way you've gotten that conclusion right is if you actually looked at the data and you didn't stare at it. And like that's a lesson I think that we should remind ourselves of also in this day and age of LLMs. Like it's very easy to come up with a flashy demo. But you got to be very critical and maybe what we got to do is come up with tools that help us. So back in the day when I was teaching this for the first time, we had static charts. One thing that's really cool about Alter is nowadays we've got these more interactive charts, right? So I can actually pan and zoom in and stuff.

But interactivity is only part of the story. And what I want to showcase today is how reactivity is actually the cherry on top that you might want as well. Uh before moving on to that though I just want to make sure any questions about this. Yeah I mean I could um just out of curiosity like um is there any intuition behind the design decision to put the plot above the code? That's interesting right like oh right I mean that's a setting. So to be clear that's a setting you can configure this.

Well, one thing that's kind of unique about MIMO uh and I'll let me explain that first. So, we're now going to go to P to part two because that's that's the bit where I can maybe explain that. So, these are three cells and like if you were doing stuff inside of Jupiter, let's say, right? Like you would run this cell, you would run the next one, you would run the next one like shift enter, you know that shortcut and you could then also maybe do something like I don't know $a - b + c$. You could have a cell like this. Um, but the thing in MMO is we can say, well, the order of the cells don't really matter.

And can I stop you for one second because you did something that is in on my mind. You have these like notebooks in like a continuous canvas or something. What is going on? Oh, I've got columns. Oh, you have columns? Yes. Okay. So, you can do that. Is that a So, that's like a MIMO thing. That's like a unique That's one of the many things, but yes, that's a setting you can add in. So the um I don't want to press the button now because the whole screen will move, but you can have like very wide cells or very thin cells. Some people like the the linear thing. I really like to have columns because then one column is typically like boilerplate setup and then the next column is like slightly more interesting stuff and then you know you can have levels of abstraction kind of logically, right? And but this also kind of falls into the category of well the order of cells doesn't necessarily matter. Uh, I could move this into another column and the code would still run. And the reason why is because we do a little, um, trick.

So, imagine that you're the notebook and you have a cell over here that says, uh, I want to calculate $a - b + c$. Then I know that I need these variables. And that means I got to start looking for cells that have those variables. And I'm just going to attach some names here. So, this is cell one. This is cell two. Roman numerals. This is cell three, right? Four. What I could do is I could say, look, this is kind of like a DAG. I have cell four, and in order for that to run, I need this cell to be around. I need this other cell to be around, and I need this other cell to be around. And basically, before anything runs inside of a there's this pass over all the code where we try to figure out in what order the cells have to run. Because if we know it up front, then we can also do some clever things. We can say things like, oh, um, let's say the DAG was a bit more complex. Let's say there's a node over here as well, maybe another one over here. Then we could do things like, hey, if I make a change to this one cell over here, then these cells over here don't have to run again. We can kind of be a bit picky about that, and we can save some compute.

And one interesting consequence of this is that it's also fully reactive. So I can change this one cell over here, and then the rest can just automatically update without me manually having to run the cells. Now to be clear, you don't always want that because if we're dealing with a big compute thing like giant pytorch loop or big data frame operation, then you actually want to have manual control. So what you can do is at the bottom over here, we have this on cell change auto run, you can change that to become lazy.

And then if you were to make a change to a cell and maybe run that, then you can see the color of the cell changes to indicate that this cell um needs to be manually fired off because a uh because it's dependent on a cell that just changed. And this way you can manually change all the cells if you want to. What does the on module change mean? On module change. Yeah. Oh. Uh, if I have like a Python file uh locally that's like u uh you know a module that I'm building or like a package that I'm working on and I'm importing that inside of this notebook.

Okay. Watches the import. Yeah. Got it. The and then sometimes you want that, sometimes you don't, but we we basically give you control here. And the whole like the main use case I got to say for this really is the whole oh I have like a this one cell that's like a big heavy compute thing. I don't want that to automatically trigger all the time. And maybe for some workloads, it's actually a lot better to just do that stuff manually. That's mainly what the setting is for.

Um, but one cheeky thing that's kind of interesting at this point, like if we understand that we we do this forward pass over all the cells and we try to figure out what order things have to run, there is this other thing that we could do. We could also introduce some UI elements. So I could say something like, oh, there's a slider here. So it's like starting value and let's steps of one something like that right I can introduce a slider so that thing can slide around and I can now say oh that value B over there uh that's equal to the value of that slider and now as I move this slider around we can also see that this value over here updates and if you squint your eyes you'll notice something over here change um when I hover over you can see how long each cell took to run but as I move the slide around you can also kind of see the order of So when I trigger this cell, this cell will update. Afterwards, this cell will update. So this whole reactive thing is basically happening on your behalf.

And this is kind of special, right? Like the fact that these cells can run automatically. Uh that is definitely not something that I was used to from the Jupyter way of thinking about notebooks. But this is if you think about what makes MIMO special, a lot of it really does boil down to this. The fact that we do this forward pass that we figure out the DAG. There's um it's a kind of a nifty UI here. It's like explore dependencies and like there's lots of cells in this notebook. So, it's kind of big, right? But you actually have a map of like all the cells and what stuff depends on the cells and where we do the import. So, you could also explore that if you wanted to.

I see. I have a lot of questions, but I don't want to Well, this might actually So, this might be a good time to ask them actually because I'm going to go into an example after. Okay. cuz I did saw you tweet this one time that there's a consequence that you're not allowed to do this. So you have some library mo HTML and you can't do this. And I I remember you tweeting about this, right? Yeah. Yeah, that that one uh that one caught me pretty but I think that's an okay compromise. I mean you know um well there's there's a reason for it. Namely that um this a is a global and that makes it easy for us to figure out like oh everything's kind of global. It's explicitly defined and therefore we can define the order of the cells. But there's al also like another thing that's that trips people. You also can't do this h because what's the value of a in this cell?

Do we have to assume that all of these updates happened or is this meant to be running after? So in place variable um updates is something that we actually uh prefer you not do because it makes us makes it very hard for us to figure out the order of the cells are. There is a benefit to this though and this is something that

definitely bit me a few times inside of Jupiter. What would happen if I were to remove this a variable the declaration right? If I were now to run this cell, an error is going to happen immediately because the representation of the cell has to also be equal to the representation in memory. And that's a thing with Jupiter that can catch a lot of people by surprise. If you accidentally delete a cell, it doesn't mean that the variable has been removed from memory.

If that makes sense. So this is also a enforcement mechanism for consistency if that makes sense. I I kind I kind of interrupted you because usually these two facts are the one thing that sort of uh confused people the most, but like now that I've now that I'm done talking, let it rip. What are your questions? Yeah, tell me more about the AI coding feature. Can it see all the cells? Can it see the markdown cells? Can it see everything?

What does it see? Does it see only the cells before it? Does it see after it? Can you tell us like more space? Yeah, so we we have all the variables, right? So these are defined. So these are like variables. We we have them in sight. Uh we also have these data sources. I've I've only gotten data frames here, but you can like hook up Postgress or iceberg or uh Red Shift. That one's new. Uh that one was added last week. Uh but data like um so we've got variables, but then data frames or tables are kind of like special citizens. And one thing you can do is you can also chat with AI. So I could do something like hey uh what are the columns in I think I called it chick data frame.

Yeah. So it's an in-memory data frame. We can just detect that. Um and this gives us a way to say oh there's a data frame there. There's a schema attached to it. Let's add that to the prompt. And this is using cloud for sonnet. And you know it can see like it's able to inspect some of that schema information basically. And on our end what we do is we basically make the prompt a bit bigger. We're able to inject like some knowledge that might be useful. That's something that we can do here. Um like a couple of output things happen here. You can add that to the notebook or you can copy it. Uh there is also a co-pilot feature you can turn on if you wanted to. Um and another thing that's also good to know if you go to there's an AI section over here where you can add your keys and all that. But one thing you can also do is like add custom rules. And basically I prefer polars over pandas these days. And whenever I make charts I really like to do that with altter. And this is just going to be appended to every prompt that you send out to an LLM. And does the co-pilot in the notebook see all the cells in the markdown? Like if you wanted to just stay in the notebook and not go to chat, let's say, I I would have to do a little bit of a ring check.

Yeah. Okay. I don't feel like doing this right now. So I think the the tab completion thing that we've got that only looks in the current I don't know for sure actually. I don't know if the So we have two settings there. Um so one thing I could do is shift command E. There is this generate with AI element that we've got here and this one definitely works only on this one's cell I think but I could be wrong on that. I would have to check I need to check the the code for that to be honest. I do know for sure that this one actually looks at the entire notebook and you're able to explicitly pass on schemas.

Got it. Um I don't know for sure about the tab like this is like GitHub copilot tab completion stuff that you can get within a cell. I just don't know for sure if we how much we look outside of a cell. That's something I I prefer

to be right than wrong, so I'm not going to answer that question. No worries. No worries. Um I understand. Um can you tell me more about the storage format of the MIMO notebook?

Like how is it stored? And then it is a Python file. Okay. So, uh one benefit of having a Python file is the git story is just a kind of a bit nicer. Like it's a Python file, so you don't have to worry about the JSON stuff. Another thing is that you've got the cool UV benefits. So anything any benefit that happens to like normal Python files we also get. So you can add this um I think it's like some it's a pep. So pip can also deal with this but you can add this basically to uh the start of your file. You can say what version of Python it needs and what dependencies we want. And this makes the notebook pretty much self-contained. Like you can you can run this in a sandbox command. You do `mimo run sandbox` or `mimo edit sandbox` and it can just install the dependencies right on the fly. The thing that's interesting if you look at the cells um you can actually see that this is a cell that needs a couple of variables and you can also see sometimes that a cell returns a variable and this is the basically the the reactive uh graph thing in action as well. So whenever you make an edit and we save it, we have a decorated function where oh this cell needs polars around oh and this cell emits a chicken data frame and every time we decorate one of these functions, we check like what's the input signature, what's the output signature and that way we can also construct the order in which all these functions have to run.

Another benefit of having it in a Python file like this is it's kind of like a fast API app, right? So we decorate everything and um there's like we've also got SQL cells by the way and markdown cells and like we've we've got a lot of stuff but at the end you aren't able to do if name equals dunder domain app run. So you could also choose to run this notebook by saying look just python name of notebook file and it'll just run as a python script. So what you could theoretically do is you could like I don't know if you like I prefer typer but you can also define a click app in here and then your notebook can also have a self-contained command line application that is still in a python file. So you can also use um MIMO inside a CI if you really wanted to.

Um what happens to the outputs like let's say visualization something like that. Let's say you work on so we don't store them. That's the short story. We don't store them but what you can of course do is we do have uh you can download um you can download this stuff as markdown or as HTML. So if you have static outputs uh that can definitely be nicely stored. Another thing you could also do is we have piodide nowadays. So what you could theoretically do is the outputs are still not store but you could also say well assuming you don't have any weird rust dependencies in here that pi can't handle you can also export this notebook to a static HTML file which you can then host on GitHub pages. So for lots of my new packages I basically host this notebook on GitHub pages and that can also be like an export target if you want. If you uh if you export a Marimo notebook to like let's say let's say you have a report that you made all kinds of visualizations and whatever you investigated some data you sent you want to send it to your colleagues um you know just going to open up a uh a example. Do you have to export to HTML?

Is there any way to send them the notebook with everything in it without them having to rerun it to get the same visualizations? Um, we I if I'm not mistaken, yeah, that is from the we need to run that from the command line. Like we have an export command. I think you could also uh let's just grab Oh, uh, terminal. Yeah. UMO export. Let's just grab the help. Yeah. So, you can run it like HTML would kind of be the safe bet. just um dep if

you want to export alter chart for example that'll be fine because you get everything that's just a static HTML file of some JavaScript and it's self-contained but if you need to run some Python code in order to also uh update the chart because you're connecting to some sort of API or whatever then you can hook it up to this WM back end but that's because you need some live Python to also run you could also choose to export it to uh to a Jupyter file if you really wanted to we also support the IPython notebook and then you can render that as you would normally too if you really wanted to do that. Okay. And can you go in the reverse direction like you can import no like can you import HTML into this like HTML notebook into or how do you how do you interchange um I would have to double check. So we like the main thing with Jupiter that you have to be a little bit careful with right um it could theoretically be that you have a equals in like multiple cells and that's something we don't technically support. So you will always have that right? Right? So that's something to be kind of aware of. Um but in general the the main idea uh behind this format is we don't store the actual outputs. You can do that by doing some sort of an export. But the main benefit of doing it this way is just the whole git story is just so much nicer. That's the I think the main reason why Ash the the creator um went with this style. I could be wrong on the history details there, but I think Ash is actually in the YouTube maybe as well. So, he can give details. Um, he's not in the Zoom, but yeah, maybe he's in the YouTube. Um, oh yeah, Marrimo. Oh, there you go. Uh, there's also a setting to automatically snapshot the notebook as IPython HTML as it runs.

Actually, he's in the he's in the YouTube. Sorry, what did he say? There's options to there. There's also a setting to automatically snapshot the notebook as you run. So, let's see if I can find that. So, there might be like a setting here, like a user setting. When enabled, Marimo, here you go. Um, when enabled, MIMO will periodically save this notebook in your selected format, HTML IPython notebook, to a folder_rimo, next to your notebook file.

So if I select this and then hit save and if I then go uh here then yeah we see the bulk HTML appear. So this and this is the HTML uh that contains everything. So there there is a setting that allows you to as you are working oh and then this is also an interesting thing. Um, I need to install this dependency in order to also store the IPython notebook. Let's just have that install.

Uh, there's a packages tab. That's also kind of like good to mention. So, you don't have to do the annoying exclamation point pip install package name thing. Like, we've got a a dedicated sidebar that allows you to install packages. And also, if I'm not mistaken, if I were to do something like import sklearn or um I don't know, some other like torch. If I were to do that, right? Is that also around? Okay, flask. Yeah, whenever we see an import that isn't in the virtual environment, we can automatically detect that and we can immediately add it to the virtual environment. So, that's actually something that just happened. Let's just save now. Double check the user settings. Okay, it should now save this as an IPython notebook. The dependencies are now installed. And if I were to now check here, we see that bulk.ipb got saved. Uh, yeah. And I no renderer could be found for this mime type. I think that's a VS Code thing, but uh the charts I do think would appear here if you were to add this to GitHub if I'm not mistaken.

Could be wrong. Uh but I would imagine that anything that you do get the marrimo slider stuff, it won't render of course, right? But I think if you were to add static map lip stuff here, it would be attached to the notebook. Um I think we can open this open width and then we can say as a text. Yeah. So the outputs are stored here,

right? So that's something you can see. So if you really if you really did want to store your um like visual artifacts, you could um but Mimo chooses to just kind of forgo that and have a clean Python file instead. That's the main um vibe at least that I would describe it.

Okay, great. Yeah, it was really helpful to get that uh like it is different. So the main thing is it is different than Jupiter but for me at least the main thing that kind of took me by surprise is the fact that like oh it's the reactive nature and things can automatically run and that has consequences but the moment that you gro that you can kind of infer how everything else is supposed to run but it is very much unlike Jupiter. Um if I were to like from my experience the main thing that I miss from Jupiter uh that I did think was kind of neat is you know how you could be inside of a cell and outside of it. Mhm. um those shortcuts just kind of work differently because there's not really a notion of being inside of a cell or outside of it. A good consequence of that is that I do get these columns, but because we don't necessarily have like a first cell and a last cell because it's all kind of a DAG. That's also why a lot of these keyboard shortcuts that you're used to, that's also going to be just fairly different.

Like that's the main that's the first half hour. Uh oh, my my fingers don't find the right shortcuts anymore. That's the main pain point that I experienced when I tried it the first time. What do you mean? I'm not sure I track the inside. I mean, I understand you're inside a cell and outside a cell, but would I'm not sure. It seems like you're inside a cell right now. You're inside that first cell. Oh, yeah. Yeah. But there but uh usually if so, if this were Jupiter, right, I could hit escape and then I'd be kind of outside of a cell and then um you would be able to go through all the different cells.

Let me put it this way. That was a shortcut that I used all over the place back when I used Jupiter for like quick navigation basically. Um, one thing we also have a command pallet. So like one thing you could do is you could go for like show the keyboard shortcuts and then like all these shortcuts are things that you can also manually change. But these shortcuts are like pretty different from Jupiter basically because the way that cells interact is also very different from Jupiter. So this is this is the main sort of productivity hump to get over I might say. Okay.

So, okay, having said all of this, like I've explained So, is there your main like the the reason that you personally like MIMO? Is there a certain feature that kind of, you know, takes up most of your I mean, is there a certain like flagship feature? Is it the reactivity or is it just like a whole bunch of them together? It is. It is definitely the So, you've got these reasons that pull you in and then there's like the holistic picture that makes you stay.

That's kind of the way that I would experience it. So for me, the AI thing on its own would not be a reason for me to start using MIMO. The way it does package management on its own would also not be the reason for me to use MIMO. Um the UI elements also for me, they're interesting reasons for me to try them, but it's not for me to stick around. However, when you add all of them together and when it's kind of becomes a holistic ball, then suddenly I become super duper productive. Um, I think the main moment for me, my previous employer, I was doing a live stream and I hit a bug during the live stream that was caused by I deleted a variable that was kept in memory. I had a cell that still had that variable around, but it wasn't representative of what was in memory, like something along those lines. That will just never happen to MIMO. Like there will be a big red error in my face

whenever a moment like that would occur.

So like there was this one time when MIMO actually corrected me live and that was a moment for me where I did start to think like okay that uh this is nice like I like tools that correct me while I'm in the flow of it and not like five minutes after uh when I'm trying to debug something. That makes sense. I can see already for teaching is kind of interesting already. Really I mean the column thing is really good. Yes. for for for this like what you're doing right now which is like teaching us and let me give you another thing that's great for teaching then since I have you but um before getting there I'm going to have to explain a little bit of theory because I'm about to show you like a pretty complex visualization with like a couple of steps in it. It's a trick that I've used over and over again in the past but uh let's say you've got a big data set. This is a picture of a database and we have lots of texts in there and let's for uh for keeping things simple let's say that this is a customer support data right it could be that we are interested in a very specific phenomenon um maybe we're interested in things like oh customers might be complaining about deliveries or something like that right and one thing I could do is I could type the word delivery and I could do like a basic text search and that wouldn't be a bad idea idea, but there's a lot of these like delivery adjacent words and things, right? So, instead, what you might want to do is maybe do something with vectors instead.

So, one thing you could do is you could kind of build the very standard vector lookup thing. You take every single text, turn that into an n-dimensional vector, and then you could sort of do the lookup. But the thing that I usually really like to do instead is I like to turn that highdimensional array into like a lowdimensional array, preferably two-dimensional. And then I like to turn that into like one of these big scatter charts with um basically it's a trick I think people have seen before, right? If you take lots of these texts that have clusters in them, um if you pass them uh through UAP after doing some sort of a embedding, then you tend to get this 2D chart where there's just lots and lots of different clusters basically. And if I'm done then like one thing that can happen just to clarify UMAP is a clustering al algorithm. It's a dimensionality reduction technique. It is oh dimensionality reduction. Okay.

It's been a while since yeah so it's um the main thing to think about it the main way that I like to think about it is PCA is really good at um keeping the global structure intact but that does come at the cost of local structure. Like it's kind of hard to find clusters when you do PCA. But then what UMAP does is it basically says, "Oh, but all the clusters that were there in the highdimensional space, they persist in the low dimensional space. It's just that distance between clusters no longer means anything."

That's kind of the the balance if I were to put it into words. But the thing that I'm about to show you is I'm about to show you a system where I can type a query and that is going to determine the colors that appear in this chart. And that I'm going to show you how that will just help me find data points that I'm interested in maybe annotating. And this is also like a perfect example of something that would be super hard to do if we didn't have something that was like properly reactive. So like one thing I could do is like this is the the Alter chart that's decorated. Now this looks so familiar now. I feel like you've been working on this derivative of this specific thing for a while. Even before MIMO, I've seen you hack around in this area. So I I actually So I spent years building like standalone web apps and um like Jupiter plugins even to get this workflow to work nicely for me

because it's a very powerful workflow like you solve a lot of problems if you do it this way. But if you have reactivity, it's just done. So what MIMO can do is it can if I just show the code, it's kind of a cool trick. You can give it a scatter chart from Altter and Marrimo can turn that into a selector widget. What does that mean? Well, what I'm able to do is I'm able to make like a selection here. Like I'm able to select these data points and then this widget has a value property and I can just look at what I just selected basically. And maybe I should zoom out to really show that I can select a different region here and then the data frame down below here updates.

So what I'm actually able to do is like kind of look around this embedded space. Uh something over here. What's this? This is all about online grocery items. Stuff I see over here is about club cards. So there's like clusters over here, right? And the fact that I'm able to make a selection and get an impression of what the cluster is about. That's definitely kind of nice and useful. But what I can also do is say something like, oh, I'm interested in like, I don't know, club card.

And when this cell updates, this cell can update, which means I can now make a selection. And lo and behold, this cell can also update. But there's something about that interactivity where I mean is it stacking that filter kind of like like you have that uh query. So this is just to change the color of the dots this does. I see. But the reason why I do that, there's a good reason for it is sometimes two clusters light up at the same time. And that's interesting.

Again, I don't want to stare at my data. I want to interact with it. I want to actually look at it. I want to understand what clusters exist. And there's an interesting thing like you have club cars. What if I were to go to like delivery? That's also a common theme. Oh, that's a that's a lot bigger. For example, just like being able to spot that and see that, oh, there's actually way more complaints about like deliveries than about like club cards.

Just doing a basic vector lookup and then looking at some of those results wouldn't give you this extra information, right? And it's super easy for me to just whip this together inside of a notebook. So, there's a ton of these like visual packages that I've made that I'm just not going to be maintaining going forward because they'll just become a widget inside of a Marimo notebook and we don't need the complex web app around that anymore. And again, why does that work? Well, we have reactivity.

Nice. So, that's like that feeling like um it's not necessarily upgrading the notebook. I'm kind of more upgrading my imagination. That's the thing that's making me stick. Like I kind of know that I can get quite creative with the stuff that I want to do. And again, it's just nice and reactive and it's just super nice. Any questions about this before? Can you can you scroll down a little bit? I just want to remind myself how you Okay, so you have the plot. So you have an altar plot. Yeah. So So going through the steps. Step one is query goes in. Then I've got an altar chart over here. The colors update. Then I can make a selection over here and whenever I make a selection this data like the widget that I declare here that has a value property that's a data frame and there's a text column that's what you see over here. So the only so the step is change the color make a selection I can I can also pan around by the way I can make different selections but every time I make a selection over here um basically this other cell will update as a reaction. Can you show that part where you link the altar chart to the you know so it's it's really this so this is the altar chart there is this little magic thing that we have around that so there's a

couple of libraries uh alter is one we also do the same thing for plotly you give it an altar chart and altter has this selector element but it never really had a way to give that back to python so what mimo is able to do is it is able to say like look this is still an alter chart but you're also able to make selections and when you do we have a widget over here. That widget has a property called value. So widget dot value and that is just this one data frame and that's basically just selecting the data. This is just a subset of the like the subset of the entire data frame. That's what's being selected and that's something I have over here as a as a value. But this cell knows if I want to display this this widget is a thing that I need. So then this cell goes ah I have declared widget. But then this cell says, but if I want to be able to declare widget, I need to have this variable plotter. And then I think I've got a cell somewhere.

I don't know where. Maybe down below over here. Yeah, plotter. And if I want to have plotter, what do I need? Well, you are going to want to have this transformed X thing. And eventually it says, oh, I need this query over here. So it's No, that's interesting. Did you run that cell that was all the way down there or does that matter? Like what? It it runs automatically. So it I don't have to worry about that. That's the whole that's the whole point of MIMO.

MIMO looks at the variables in the cells and just by looking at that it can kind of um it doesn't do this at runtime. It does it before running uh all the code. But the whole point really is oh u I know what variables this cell needs. Let's look for the parent. Let's keep on looking for the parent. Okay, we've mapped everything. Okay, now we can run everything once. Oh, a cell over here changes. Okay, what children do I got to update? That that is all handled for you.

Okay. Cuz like in a Jupyter notebook, you would Yeah, you definitely would have to you would have to scroll up and down, run a few cells. I've been told there is actually a plugin nowadays that can do some of that stuff. And if you build your own widgets uh with the IPython uh widget library, there is actually like there's a couple of party tricks you can definitely do. Um but this is just so much more intuitive.

Like this is a I don't have to think about it kind of a feeling. And I see you've nerd sniped a lot of people already with this fast HTML. I'm not going to spoil it, but I see that you are teasing it in the next column. Um, oh, oh, people have spotted that, have they? Um, we'll get we'll get to that. Okay, I'm talking to you. I'm I'm assuming the fast HTML crowd's going to show up. I better come prepared, right? Like do the homework. Um but anyway like this on its own is kind of a cool demo like already I like to think like okay this is pretty useful this is also good for qualitatively understanding what this data set is about basically right that's cool but why stop here like if I were to now think like what's actually useful to do like let's say okay we we know that we are interested in getting some complaints about deliveries um if I make this selection sure some of them are going to be about delivery But some of them won't be. But these examples over here are like quite proximate to examples that are about deliveries maybe. Right?

It kind of feels like we got to do some annotation here. If you want to turn this into a classification problem or something like that, right? Then maybe a good next step will be just labeling. That's where this little guy is going to come in immediately because um there is a standard nowadays and it's a really cool project. It's called Any

Widget. Um, any widget solves the problem of building a widget that will work in Jupiter and in Collab and in MIMO and like all those different notebook environments. And any widget is just kind of the standard that everyone else can wrap around. And this is a thing that allows you to make a widget basically. So, uh, Jupiter also has like a widget library. Um, and the way that Jupiter handles it was always a little bit different than Collab, which is always a bit different than VS Code. The whole point of this library is that there's one standard that everyone can kind of just adhere to. It has a small surface area, but what it does allow you to do is build custom widgets for yourself inside of your notebook. And who maintains is this a MIMO project or is it a third party or what's the background of it? There's this uh guy called Trevor. If you uh I used to host a podcast for my previous employer. Uh Trevor was the first guest. Trevor Mance. He did a PhD um kind of evolving around widgets like this. uh a bit more in the biomedical space. Um if I recall correctly, he is a direct colleague of mine, but it deserves to be stressed. Uh any widget is like all notebooks and that's not going to change. So yes, Trevor is working with me and my colleagues and we're definitely eager um to invest in any widget, but any widget itself will never be MIMO specific. This is across all the notebooks basically.

So if you build an any widget for MIMO, it should also work in Jupiter. That's the way to think about it. However, the the one thing I do want to mention is that um the reactive nature of MMO really lends itself well for any widgets as you might be able to imagine because what these widgets are able to do is you're actually able to change something on the JavaScript side of things and then see that uh reflected in your notebook. And maybe the best example of that um I'm just going to maybe do something down below over here. This wasn't part of the original demo, but I might as well show it. Um I made this any widget library called draw data. So from uh draw data, I'm going to import a scatter widget. Let's just zoom in a smidge.

Uh yeah, and then draw data needs to be installed. So let's install that. It's installed. Good. And now I should be able to do scatter widget. Uh wrap that with the MIMO any widget wrapper. And then this is a scatter thing. And then I should be able to just uh show you what the scatter widget looks like. Um, so this is a tool that allows you to literally draw a data set. And this is amazing for teaching because you can literally uh draw a data set as you see fit. And the cool thing with that scatter widget is you can actually um data as pandas if I recall correctly.

Yeah. So it can come out as a pandas data frame. And every time I draw more rows are being added to the data frame. So that's super neat. But what you're seeing here is I'm I'm doing something in JavaScript land and somehow that finds its way back into Python. So did you define this widget with Python or JavaScript? Both. Okay. So if you look at the um any widget example on the main docs, I think it does a really good it does a good job of explaining it. But let me just uh can I zoom in? Yeah, this zoom in and uh change the color. Notice by the way that I'm uh doing the doodling thing again. Everyone buy the drawing tablet. It is amazing. So this is how you would define an any widget.

Um you have to import these two libraries. One of them is called any widget. The other one is called traitlets. Traitlets can be seen as a library that is able to bind variables such that JavaScript and Python can sort of communicate. Um and then there's this one value that I attached to this object. And then this is the basically ESM JavaScript string that has this one function called render. It needs to accept a model. The model represents all the variables uh that you declare from Python and that are shared between the JavaScript side of things in

Python. And this element is basically the DOM element that it needs to attach onto. So in order to write these, you are going to benefit a lot from being able to do a bit of JavaScript. But what you can also do is have someone like Vincent who actually doesn't mind writing JavaScript. and I will build these widgets for you such that you can just use Python inside of a notebook that you like and you can still benefit from um like having this interactivity as part of your workflow basically.

But the way it works it's not it's actually not that intimidating. So if I just go over like all the things that I that are listed here, right? So okay, what's the first thing that happens inside of the JavaScript thing? Well, we declare a function where we're able to get the value which is this trait lit that we've defined inside of Python. After that we're going to create a button element. Um, we're going to say, hey, there's a count attached there.

We're going to add an event afterwards. Um, like whenever we do something in JavaScript, that's what this this event here does. But we also have something clever here where we say like, oh, if ever the value changes from the Python side, this is how the JavaScript should respond. And that's about it. So, it's basically what should happen if JavaScript makes a change to Python, what should happen if Python makes a change for JavaScript, and then you've got a button that has state. Uh, and that's the way uh it works. Cool.

Um, any widget is the bee's knees again, just like the drawing tablet. I don't understand why I'm the only guy making these widgets. There's a small crowd of people actually building these widgets. It's a cool Discord, by the way. Like, do definitely join. And does it only work is it only appropriate for, you know, notebooks or can it can you put it in your web app in just the same way or it's no? So, um I've I mean Trevor would know the fine details there, but it's a different interop basically. Um because Jupiter has like a certain way that it expects variables to just kind of be there and uh it's I've not tried getting it to work inside of a proper web app.

Um like I I'm 99% certain it's going to be a painful journey and it's going to be way easier to just use something like fast HTML instead if you want to have this kind of interactivity unlike a normal website. But the goal here really is to give you features for your notebook experience also. So it's kind of um um it's not the target I think for the project. Again, if Trevor is watching, he can confirm. Um but my impression is that this is really just meant as a tool for you to build tools inside of your notebook. Fun fact though, um there are actually a couple of very popular libraries that actually use this under the hood now. And if you go to the community tab, um, Alter actually uses this under the hood, which is like a thing that not not everyone knows about. Uh, there are also these other libraries, um, more to do with like mosaic. Um, there's also some like physics things in here, by the way. Um, but there are like pi opsplot is also like a really cool plotting library, by the way. But there's like a small community and there's definitely like very cool widgets um, for sure. Uh but it's kind of one of these libraries that are meant for uh package makers. So it's not that wildly known uh amongst the user group if that makes sense.

Yeah. But it is a super cool project and if you're interested in learning JavaScript, this is probably the most fun you'll get like building your own widgets. Um if people are interested in talking to me afterwards, I am actually doing a bit of prompt engineering right now to see if we can get these widgets automatically generated on the fly. like you the dream that I've got is that in a couple of months you'll be able to go into MIMO right and you'll

be able to sort of start typing here something like hey build me this widget because I want to do this kind of research and the widget will just be generated for you inside of the notebook. So what's the what's the devx of that widget building experience since you are writing JavaScript inside a string like oh you can also have you can also have it in a file and then you can have and you can just say like hey there's a static folder here Python path library open that file put the string in here that you can also do. Is it a nice way to debug and let's say kind of play with it like really quickly when you're writing the JavaScript, you know? I mean the main um to me um it all became easier when I learned of this one shortcut shift command C because what you're able to do is you're able to click an element click and then DevTools will immediately bring you to that one element that you're interested in debugging.

like it's the the annoying thing was figuring out like how do I select that one element inside of a Jupyter notebook and I don't want to click around here forever. That that used to be the painful bit, but then once I got that figured out, it was pretty good. I will say you can also get quite far with vibe coding those kinds of things. It's just that um if you want to debug it, it does help if you've done like D3 before, let's say. Oh my goodness. D3. Okay.

Yeah. Know I like I like that that one is a steep learning curve. Yes, it is. But at the same time, it was also the reason that I started to learn programming. Without that library, I probably would never picked it up. Yeah, it is super cool. No doubt. Yeah. Um, so the but like one observation I do want to maybe make here is like, hey, this is kind of interesting. Like when we're talking about interactivity, MIMO comes with widgets, but nothing is stopping you from making your own. And the way that I've set up this notebook is what I can do is I can actually maybe make a selection over here. This is something that I might want to annotate.

And then uh here we are to the widget that I sort of uh want to really highlight here. Um this is an annotation widget. There are lots of ways to annotate data. Um sometimes you want to make selections in an image. Sometimes you want to make selections for spans and text. Like there's lots of different ways to do it, but nine times out of 10 you can boil it down to yes, no, and skip. like it's kind of a binary situation that you can often boil it down to because sometimes it's a classification. Can I can I apply this label? Yes. No. Sometimes it's oh there's these two variations. Which ones do I like best? Uh sometimes it's oh I'm doing retrieval. Is this example a good thing that fits this query? Like so the way that this thing is set up is basically you have to generate a card here in the middle and then you can say yes, no or skip. And you can also go back if you wanted to. The way that you generate this is up to you. The only thing you got to do is give me a string that really looks like it's HTML that I can render in here. So, if you want to use fast HTML for that, totally be my guest. But the thing you got to do is you got to give it a list of examples and a function that knows how to render each example in here.

Then I can render that in the middle and the widget around it will from there provide you keyboard shortcuts uh which will also be displayed. And I've also built it in such a way that the gamepad API is also supported. And the thing that I'm then able to do is I'm able to sort of click around and say like yes, no, yes, no. I'm able to go back as well. But the really cool thing is sometimes you also want to add context.

And if you're using Chrome, uh Chrome has a web API for speech detection and it's free. So what I'm also able to do is hit this button and then I can start talking and then oh this is now a note that will be attached this one annotation and I'm done talking now and I can go to maybe next and if I were to go previous I can see the note again and I can sort of there's lots of these examples I can imagine like especially in LLM land where oh the bot made a mistake here but for downstream tasks I got to explain why this is a specific kind of mistake right um so for that use case I also make it easy for you to just add a note um if you look at the annotations at the bottom here. So, can I ask you a question about this annotation interface? Um, what part of this annotation widget, this card, I guess, with the buttons like is that all defined in fast HTML or what? Up to you.

What in this demo? What was the render? This render function. So, this render function is fast HTML. So, I wrote kind of my own thing called Mo HTML. It does a few things that I like. Uh but any like you can also use Ginga. I don't really care. I will say fast HTML does lend itself very nicely for this though. But um you have an example that goes in um and what I want to render is some sort of span on top uh that tells the user what the label is, what the task is basically. And then at the bottom I have this example text that I just want to show. And there's like a couple of Tailwind classes that I attach. But the way that you render this, um, it has to be a function and it's up to you to choose what you like. But fast HTML does lend itself very nicely for this, especially because I think Isaac is in here. He made monster UI and there's like a couple of these packages that come with nice UI elements. Those can just be dunked in here and they just work. Can you scroll up again? I'm trying to just uh locate where this label appropriate thing is and the other text. Sure. In the UI. Yeah. So, um, so let's let's first start here. So, we've got those examples, right? Yeah. Yeah.

And let me let me maybe start there. And again, let's just look at the first. Oh, I see it now. It's part of that. Okay, I see it's underlined and it has the But so, okay, that that text part is basically your that's the renderer and then all the buttons and stuff are kind of your widget that you already made that we plug into. Yeah. So, so the thing I I well the thing that I built around it is uh Whoops. The thing I built around it is this progress bar, these buttons, the notes thing at the bottom, and it's up to you to populate this with something meaningful. I see.

And and how do we do that? Well, you supply a function that can do it. And as far as I'm concerned, fast HTML lends itself quite well for that. Mo, I really like this. This is like a really good trade-off or this is like a kind of a good design pattern. This is I think you would have to be really annotating a lot of data to know but I I think this is like I've built annotation software before so it's not my first forego edit but the main thing that I like all the things that you just said are true but there's like this one extra thing I want to add here you can still do Python around it so if you want to do deduplication up front send it to a specific database uh do user stuff you're in a notebook and you're free to do whatever That's the thing that I really like about this approach mostly. Like sure, the UI is kind of like nice and all that and the game pad is awesome, but the main thing that's really nice is it doesn't get in the way. You just add a widget to your notebook and oh, it's pretty fun to just get started annotating and you can be done in 10 minutes and in an hour you have 500 examples. I don't know how fast people like to annotate, but it doesn't break your flow. That's the main thing I like about this thing. You don't have to start up a new app. You can just sort of add a widget and your notebook can now also annotate data.

What I really want to get at staring at their data and to start looking at it is to make it fun and easy. And I like to

think that this is indeed a very nice balance. Um the thing I made earlier, by the way, like this is this is I think a widget that combines a few elements. Um the speech to text thing. There's also a library called Mo Talk that just has a widget that only does speech to text.

There's this other widget uh called Mopad that is doing the game pad stuff. Um, like there's also widgets just for that, but don't sleep on these widgets in general. Like other things that you could do, um, I don't know if people played around with the Gemini API, but the Gemini API actually allows for drawings and like images and stuff. So, I made this paint widget, but this paint widget is actually like pretty dang neat. Like if you want to demo that I've done that you can do yourself as well especially if you have a drawing tablet like this uh you can actually draw uh a DAG like this and then tell Gemini like hey what's the JSON for this and it will just go ahead and give it to you. Uh I'm building another tool like this built on TL draw which has like proper vector images and you can draw proper diagrams in that and then hand it off to Gemini and they can actually do mechanical engineering tasks for you.

And again like why stop there? Uh, we can paint, but what we can also do if you're into like PyTorch and stuff, I can also come up with widgets for like charts that live update. So, I can have a cell over here that has a chart and I can have another cell that updates that chart in line because I can change the state of the widget. And that's something you can now totally also have in your notebook. You can have these cells that sort of asynchronously update as while some training is happening.

And on top of that, and that's kind of I think a nice place to maybe end as well. Uh, hey, one question. Those annotations, um, how do you save them? Do you like do you do that? Yeah, like where do you Yeah. So, up to you. So, like one thing, uh, as I annotate, right, the cell over here updates and there's a list of items. You can write some logic here that says every time I don't know I get like um I have like 10 annotations or maybe 20 and then 30 etc.

then I'm going to send the request to some back end to store it in a server. Uh you can also write a little bit of SQL alchemy here if you really wanted to. That's kind of up to you. You can store it into a file as well. It depends on the that sort of stuff usually depends on what you like to do as a team but you can write Python around that to take care of it. Okay. this label widget get annotations when what is the event that triggers that to run.

So this thing triggers every every time you make an annotation. So you could store every single uh thing that goes in. You can also choose to batch this. Okay. And you can um Yeah. So you can do something like if uh length of this thing mod 10, right? then the cell will still run every single time, but it's not going to cause an event to happen downstream if you were to like add a mod here. Like you can do stuff like this.

But again, like this is like um I remember working at this company and then the community would ask us things like oh could you add support for MongoDB and like oh could you add support for Postgress and oh could you add support for Unicorn DB and like you kind of get like all these crazy requests of things you would like to add. So the main thing that I would recommend people do here is just write Python. You can figure out. So is

there something like get annotation but uh singular like get and like you know just gives you the annotation uh just most the most recent one you mean? Yeah. You would just do it like this. Okay. I mean um I made this thing like two days ago so it's pretty fresh. You can definitely like use it for a bit and then tell Vincent to maybe consider a feature.

Right. That's also something that can happen. Like the main thing I want to do here is like this is also me presenting this for the first time to the outside world. I'm also here eager to get feedback, right? So like if um there's a preference to maybe uh only have the latest thing or like something like that, we can always add that to the library. That's that that's definitely reasonable. Got it. Okay. I mean this is like isn't like okay this is not you saying this is our annotations or like our widgets library that we've you know thought about for extremely long time. This is like, hey, you can build whatever you want in Mimo and kind of look look what I built. You built this thing for annotations and you can build something too if you want.

Yeah. I mean, it's kind of like Legos. That's the best explanation I would maybe put here. Like, um, MIMO defines how things can click together. And what I really like to do is come up with fancy new Lego bricks that you can maybe use, right? And a lot of it is because, oh, there's a new Gemini API. Oh, it does stuff with images. Oh, I need an image input. Oh, it doesn't exist yet. Okay, let's build one. Um, so a lot of the stuff that I do, like I'm uh I'm I'm technically the first hire over at Mimo.

So I am technically like a developer, although a lot of stuff that I do is a bit more based around growth. And the way that I like to maybe do that is I also do the YouTube thing, of course. But a lot of it is also just come up with widgets to see if this can actually serve a community. So right now I'm very much into the, you know, data quality. That's definitely like my background. I want to make sure we got good widgets for that.

um depending on what we got to focus on this summer. I don't know, maybe we do some stuff for maps, right? Like I can definitely see that kind of becoming a nice little thing to maybe work on. Um so we'll see like what has priority. We're we're a small team, but we ship like three releases a week. Um four if you include these widgets. Uh but the thing that I'm trying to do is have like more widgets out there in the ecosystem. That's definitely kind of like the thing I really try to do over at MIMO, which is also like every month I tend to release two of these widgets.

Okay. Uh there's been a lot of comments on the uh Zoom that you have a YouTube channel where you keep sharing different ways of using MIMO and it's highly recommended. So it gets compliments online and we are not we we try to get we try to if we make a video we try to not be too hyp about it and make it worth your time. We only try to show stuff that's actually kind of useful. That's definitely a goal that we have in mind. But yes um definitely check us out. Uh we also have a really good discord by the way. Feel free to check that out as well. Um I do want to end on like a final thing um because that's something that people sometimes forget like a thing that you can also do just as a programmer right uh uh let's make a a class like I don't know um fuar I don't know we're going to initialize it just like you would normally right and um I don't know you would do like self thing equals thing let's maybe add a thing in here.

Fine. And then if I were to do something like a fooar, this is what the representation looks like. And this comes from the era where we are doing things inside of a terminal. And that's fine. But for Pete's sakes, we're in a browser. We can do super awesome things in here. Like what one thing you could do is maybe consider doing some interactive HTML there. So like one uh trick that you can do that's marma specific is you can add this uh display method.

Right. And then I made this one library called MoHTML. Like one thing that's kind of cool about it is you're able to do like I can add some text in here. Hello. I could do like a class uh font bold if I'm not mistaken. Right. If I were now to run this Oh, like I can display a rich HTML thing in here that kind of belongs nicely uh to this class. And the best example. So there's also the uh dunder. Yeah. Like HTML thing and you know. Yeah. So there's this repper HTML is also something that exists. Um this will work across Jupiter and uh MIMO. The only thing like the reason I bring the display up is because this is Mimo specific but Jupiter does not look for this one but we do. So if you want to make something as Mimo specific, this is the way to do it. Um this is also definitely something that you could do though. So if I were to Why would you want to make something maro specific?

Like what's an example? Um so one thing that we've got so MMO app meta and then mode right so this is some so right now I have code that I can write that detects that right now this app is running in edit mode right but I can also run in app mode like I can drag I need to zoom out here but I can drag uh the outputs of cells into a dashboard if that makes sense and this in doing this uh this is like the the widget editor that memory mode gives you. So you can make these boxes smaller or bigger, but now you're running in app mode. Oh, maybe the way that you display something needs to be different if you're running it from uh edit mode or app mode, let's say. Okay, that that distinction is like a very specific thing, right?

And and there will always be things that are just really MIMO specific. So for that, we have our own little verb, but it should be stressed. Um, like nine times out of 10, uh, I would already be super happy if more people would just do this. Yeah. Yeah, it makes sense. Browser, do it. Do something cool with it. Because if you do that, then you can, I don't know, maybe make a dice library where, oh, I have a dice with eight sides and it just looks like a chart with all the probability values that are in there.

And what's kind of cute then is if you were to maybe add you can add two dice together, hook up a slider to it. And this is just your your Python code just becomes so much more visual, right? And it kind of feels like more people should maybe get that as well. Like this is this is also like something I would love to see more libraries do more of like think more in terms of like hey we're building tools uh for a notebook like um can you tell us a little bit more app about app mode like oh sure is that can you how far can you go with that like in the sense can you deploy basically web applications this way do you see people putting op in front of it and stuff you know how does it what so we Well, so like uh there is a demo on our GitHub where there's a fast API app in front of this and then fast API handles the off for you. Um the because MIMO is built on top of Starlet. So like if you really want to do a proper integration, it helps if you're building something that's like Starllet compatible. Um I will say there's like a few ways to deploy things inside of Maro. So one way you could do it is you could say, "Oh, let's deploy this to Wom, right?" And then you can put O in front of that.

Another thing that you could also do is say, "No, let's deploy on like a proper Python back end because we need like GPU support or what have you, right?" Um, if you want to take a notebook and run it in edit mode, you would call from the command line `Mimoedit notebook file`. If you wanted to do something like, oh, actually run it as a web app, you would do `MIMO run` instead. So that's the that's the main like from the developer experience, that's the main distinction. Okay.

Yeah, it looks really cool. I need to give it another go. This is like a a really targeted way to snipe me into Mimo that you have come up with right now. Trust me when I say it wasn't just you that got nerd sniped by the game pad by the way. Um but um like I'm also just still kind of scratching the surface of what you can do with remotes. Uh like we honestly we are really good at making releases to the extent that like a lot of our team members don't always know about all the features just because we ship so much stuff. That's actually we had a small little meeting about that the other day. Um definitely expect us to keep on adding features. Um you can also definitely expect us to build something cool over the summer and that we're probably hopefully going to announce in September. So that's also something to look forward to. But the main thing that I really hope to get across here um these are all just tools, right? But please like expand like it's we're not upgrading a notebook here. We're trying to upgrade your imagination as well. So if you take anything away from this, it's really not so much like use Marrimo, use Marrimo.

It's it's really cool. Like use it. But I want people to dream a little bit more. Like that's that's more kind of the point. uh especially because we're like pretty soon we might be able to generate these widgets by talking to an LLM, right? And that might become a really big part of the future of how you work in a notebook. But in order to get there, I do need people to sort of let the idea of what the old way of working in a notebook was, I need I needed to have people to let that go a little bit because what Brett precisely what Bret Brett Victor said in that great talk. um we don't really know what the future's going to be yet. So we're going to try a whole bunch of stuff and we can be really creative but only if we allow ourselves to be right. So that that's the main sort of Yeah. No, that's great. That's uh end on Brett Victor. Can't go wrong. Um can you tell me a little bit before we go? Um how does Mimo make money? Like can we you know is there a paid aspect? uh some kind of hosted version something. How does it how does it go? So there's uh nothing I can uh let me put it this way.

I can't announce anything on that front just yet. But uh one thing I can say is historically there was um it was two people before and they got a grant which allowed them to work for like I think a good year and a half on MIMO. Uh then at some point the choice was made to go with the VC route. So this is seed funding that we're at. Um it was a round led by some pretty big chiefs. Uh Anthony from Kaggle it was in there. I think Wes McKinnon there like big of big group of old school CTOs that wanted to do a seed round. So we're in that. The focus right now uh within the group is definitely like let's just make open source really really good and let's see if we can get like a good community around it. That's the like I would argue primary goal. You can definitely imagine there's going to be at some point a version of MIMO uh that does some enterprisry stuff as well. Uh and that's something that will definitely be explored. But what you're seeing right here and right now, this is I believe MIT licensed. It's on GitHub.

This is not going to go away any anytime soon. Like we're not going to do a rock poll on something that's open

source right now. That's not going to happen. All right. Sounds great. Um this is really great. I will you know visit if you don't mind sharing the notebook or github wherever it's located if you um if you go to the okay go ahead yeah so if you go to coning/mo label uh yeah so you can actually play with this thing on github pages because it is wom compatible but the notebook that I'm using right now bulk.py Pi is on the GitHub repo itself. The data sets that you need are also self-contained in there. Okay. Mo label.

All the all the plugins that I make on beha that are kind of like MIMO related, they all start with Mo. So there's Mo HTML, there's Mopad, MO talk, uh Mo Paint, uh MO label. Nice. All right. Well, thank you so much for coming. Um I will post notes with all these links including you know the drawing pad and all everyone had wants to know what everyone just wants the link they just want to buy it so I will figure it out and I will mo paint mo paint you can go to the you can also go here and immediately play it with mo paint it is it's going to load it's going to work from GitHub action oh what I mean is the drawing the your your hardware oh that thing yeah okay this there's a calm code okay so if you go to calm code.io io. You scroll all the way at the bottom and then there's this meta uh category and there's content that is basically a small course that I made five years ago. All of it is pretty relevant still just a different drawing tablet. Um oh that's your uh um that's your website com code. It's one of them.

Uh I have a few at this point but uh like com code is a learning thing I made over co. It is probably the cheapest data camp alternative you will ever see. It has like 106 courses about all sorts of Python and data topics. If people want to know more about like stuff that I do like day-to-day, I also have a blog as well as a a Substack if people are interested. Um, and I guess like the final like there's two things. Yeah, a couple of personal plugs actually. I have a personal YouTube channel where I tend to review these ergonomic keyboards. Um, the one thing I want to say to that is the ergonomic keyboard doesn't fix everything. Um, but I've met a couple of people that really ruined their wrists in their career. And I'm super happy that I'm not one of them. I was early enough. Uh, do take that stuff serious. And if you want to have honest reviews, definitely go here. But if you ever feel like wrist pain, take that serious. Do exercises, talk to a physiotherapist. Don't ignore it. It could risk a career. Like that. That is a thing I do want to maybe mention explicitly. Um, yeah, I know people like that for sure. Yeah. And you just want to prevent it. Uh, there's a couple of really good boards that help you do that. This is one of them. But um pull reviews can be found over here. And then a final thing um in this age of like LLMs and stuff, I'm also interested in writing software that's just a little bit more human. And this is more like personal project territory. And this is more like I'm sure some people find this interesting, but one thing I made a while ago is this service called Dear Me email. The way it works is you can send yourself an email and you get it back in 30 days and that's it. It's kind of like a way to reflect if that makes sense.

Um, I made this a while ago. I I gave it a little boost now because I want to sort of get back into this human uh software a bit more. I know of at least one guy that was dealing with alcoholism that ended up sending himself emails and it actually ended up helping a whole lot because it's really hard to be critical against yourself. Like if you yourself are reminding yourself of like goals that you've set and you're failing, something about that is really impactful. So, if that sounds interesting, uh definitely come talk to me about that as well. And another thing I'm currently working on, um, I've noticed that it's very tempting to keep on asking chat GPT for things to the extent that like I actually want to know things myself as well, like from the top of my mind. So I'm also exploring um like a variation of Anki um that's just, you know, live and it's free and it's out there. Um, next

week I'm also going to add gamepad support for this. Um, but I'm definitely interested also like from the point of view from a of a developer.

It's really great that we have these LLMs at our disposal, but I would hate it if you become so dependent on it that at some point it becomes kind of a learned helplessness kind of a thing. So, if people are interested in maybe exploring that with me as well, definitely reach out to me on socials, too. These these are like side projects. Does this um does this like project potentially pull from your existing conversation history to find repeated things or something or where you I might go there. I might go there. For now though, I just the thing I want to focus on now is just that oh, if you're on the bus, the mobile experience is just nice and uh oh, if you want to use a game pad, that is just kind of nice. Step two would be um how do you generate these flash cards to start with? But part of me is also wondering like maybe the act of building the flash cards yourself and being very conscious about that is also good for like deliberate practice if that makes sense. So, I'm I'm definitely eager to explore the LLM generates flashc card thing later, but for now, I just want to iron out lots of other things. It's more like, hey, do other people also have this feeling of I'm becoming too dependent on these LLMs. Would it be good to maybe invest in a few apps like this? If people are interested in that, then definitely also reach out to me because that's like uh definitely territory that I'm personally exploring right now. Okay. Yeah, we should talk um you should also talk to Isaac. Isaac and I work with a company called Yeah, I read about that. Okay. Oh, yeah. I read about that. Uh, no, like we we will we will talk about that later. There there's like one there's like one thing I'm wondering is if I want to make like a remember like like a car flash card thing that's specific for developers or if I want to keep it general because I'm I'm I'm uh I'm having the impression that actually it makes sense to maybe do something bespoke for developers because especially if you put like an LM in the mix um you can actually make something that's a bit more bespoke but we'll talk about that later.

Okay, great. Um there's so do you have a site that has all your things on it like like in one like hey I have doing this project this other project whatever the I once a month I try to send an email of like hey here's stuff that I've been working on that seems pretty neat uh and also I people should blog more about prompts that they like by the way if like I learned people did yeah oh like I like I don't want to be on LinkedIn to to get signals like this I want to be like on the normal internet um so I I try to share promp prompts as well. Like today I learned thing from Simon Willis.

I just try to do as much as possible. Um yeah, I try to learn in public. That's kind of the person that I am. So if people are interested in that, I have a Substack. You can also I'm also very active on like Discords these days. So the Marima one, the Any Widget one. Uh I'm also like in the fast HTML one from time to time. Uh, but Discord is also where people can find me and I'm on Twitter and places, but um All right. Um, okay. I promise this is goodbye now.

We can actually say goodbye. Um, that was a good session. It was fun. Let's just hang out again. It was fun. Yeah. Yeah, we should totally do that. That was that was really cool. I'm going to try to hook up my note my game pad after this and I'm going to check this out. But you have successfully sniped me into Mimo. So, sweet. Mission accomplished. I can now have a cake. Uh, cool. Then everyone, thanks for listening. It was a pretty full crowd as well, so that was also very nice to see. Yeah. Yeah. A lot of people will be getting the recording, so I'm gonna be

sending it to all the students.

All right. Well, thank you. Perfect. Have a good one. All right. Thanks.