

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Jackie D'Elia: Hey everybody. This is Jackie D'Elia with another episode of Rethink.fm for you. Today, I have a very special guest, Rachel Andrew, who many of you may have heard is an expert in the new CSS Grid and is also a co-founder of Perch, which is a simple CMS application. Hi Rachel.

Rachel Andrew: Hi, good to be here.

Jackie D'Elia: Thanks for joining me. If I've left anything out and for anybody who doesn't know you, if you want to introduce yourself that would be awesome.

Rachel Andrew: Sure. Yes, I'm Rachel Andrew. I've been a web developer for, well, a very long time. I've been running my own business since 2001 as a developer and now, as mentioned, what we mainly do is Perch CMS. That's our product. I'm an Invited Expert to the W3C CSS Working Group, so my interest in CSS and the open web goes back a long way. I'm just back from Tokyo where I was at the CSS Working Group meeting talking about all things CSS. That's really what I do. It's mainly Perch, and then quite a bit of speaking and writing about CSS.

Jackie D'Elia: Starting off with the CSS Grid, how did you get involved with that and what role have you played?

Rachel Andrew: I just spotted it really. In fact, it all started, I was speaking at a conference and I can't remember what about. There was workshops as part of the conference. Bert Bos, from the W3C, was doing a workshop in French, my French isn't great, but he was doing a workshop in French and showing some kind of experimental layout stuff. One of those things was a forerunner to CSS Grid. I saw this and thought, this is something we want. Then I realized that actually there was a spec had come through from Microsoft, and there was this early version of the Grid spec was developed by a team at Microsoft. It was implemented into IE10. I got my hands on that and I started building things and saying, "Yeah. This is a really good thing. This is something we want."

Really it just started from there, that I was very keen to promote it and to write about it and to get people talking about it, to make sure that it was something that got implemented into other browsers and didn't just end up being something that ... Some sort of IE experiment. That's really where it all came from is I was just very keen ... I'm very aware of the fact that web developers have got quite a lot of power in terms of what ends up in browsers, if we talk and write about it and make some noise. I just wanted to start doing that, to champion this spec that I thought really was interesting and was going to solve a lot of problems.

Jackie D'Elia: CSS Grid had its beginnings with Internet Explorer?

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Rachel Andrew: That's right, yeah.

Jackie D'Elia: Wow. That is very ... It's not something I would have thought given the history with Internet Explorer.

Rachel Andrew: It's surprising. I mean the team now with Edge and really going quite a long way back, there's been some great stuff coming out of Microsoft and coming out of that team. There are various things that we can only test in IE or Edge. It wasn't just Grid. For a long time, they were the only people who had any kind of implementation of Grid layout in production browser. Yeah, you maybe surprised. There's all sorts of stuff that shows up, and they do some great work over there.

Jackie D'Elia: Wow, okay. How does CSS Flexbox relate to this?

Rachel Andrew: Right. Flexbox is designed for a slightly different thing. The main difference is that Flexbox is for one dimensional layout. That means that you're laying things out as a row or as a column, but not both at the same time although you can wrap flex items. If you're working in rows, each row essentially becomes its own flex container, so space distribution. If you try and line things up, it's only going to happen across that row.

It's not going to happen in columns as well, which is why if you've got an odd number of items on a row, they're just going to stretch out right across the row. They're not going to line up. Grid was designed to be two dimensional, which means that you can line things up in rows and columns and also do things like have negative space in your design, because you don't have to fill every cell on the grid. Whereas Flexbox is just going to fill up all the area.

Jackie D'Elia: For developers and designers that haven't used either yet, and most likely many of them probably have not used CSS Grid yet but may have started experimenting with Flexbox, what design elements are best solved by each? For somebody who is thinking, okay, where would I want to use Flexbox in a design? Where would I want to take a look at using CSS Grid?

Rachel Andrew: Right. With Flexbox, if you've got some things that you just want to, say, space out, navigation is a good example of that. You've got a navigation list and you just want there to be equal space between all of the items. You want to let the content control how big it is, and you want some nice bits of space between the items that's all even, so it all looks nice and tidy. That's a good Flexbox use case.

Transcript  
[rethink.fm](https://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

With Grid, what you're doing is you're defining a grid and then you're putting the items into that grid. In the case of your navigation, if you had a very long navigation point, what it would get put inside maybe a smaller grid track which might mean it wraps onto two lines, for example. Whereas Flexbox would let that one take as much space as it needed, and then just manage the space distribution. It's the other way around. Grid works from the outside in. You define a grid, you put stuff into the grid. Whereas Flexbox, you're working from the content out. You're saying, "I've got some items here. I just want them to manage their space in between each other and stretch out," or whatever.

Jackie D'Elia: Okay. You have a website that you created called GridbyExample.com.

Rachel Andrew: Yeah.

Jackie D'Elia: You have some great videos on there, so if anybody's interested in learning more about CSS Grid and how to implement it, those videos, I found them very helpful. I watched them last month, I think, that was the first time that I've actually started to look at it.

Rachel Andrew: Yeah. I think actually seeing this stuff, and I've also got a whole load of small worked examples that link to CodePens. You can go and fork those and have a play around. It's a really big spec, but the actual, individual elements of it are quite simple. It's really just starting with the small parts of the spec and playing around with it and seeing how it works, and then seeing what other things you need to know to solve the problems you've got. Yeah, working with small examples rather than trying to learn the whole thing at once is definitely the way to go.

Jackie D'Elia: How does this fit in with frameworks like Bootstrap and Foundation and transitioning from using those for your grid to using something like CSS Grid as browser support fills all out?

Rachel Andrew: I think the first thing really is that there's a real mindset change between using a framework or, in fact, even using your own framework that you've developed, that maybe a float based kind of framework, and then using Grid. Everything that comes before Grid, you have to target the item itself. You have to say, "Right. This block is going to take up five columns," in this grid that you've imagined. You don't actually have a grid with previous layout methods. You're faking that by adding rules to items in the design to say, "The grid is this big. A grid column is this big and you take up five of them. I'm going to give you a width that is the amount that five columns would be."

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Whereas with Grid, you define a grid and you don't have to define anything on the item at all or you might just say how many it's going to span. You don't put any widths on the item, because that's managed by the grid track itself and the sizing applied to the grid. I think there's quite a shift in thinking, to start with, to get away from thinking, I need to be targeting the item in my layout and saying, "How big are you?" When actually with Grid, you're saying, "Well here's a space. The item's going to go in it and it's going to be as big as that space."

Jackie D'Elia: That seems like it would make it much easier to work with responsive designs and getting things to adjust automatically without having to specifically go in and change widths.

Rachel Andrew: Yeah, exactly. Because you can do it all on the container, so if something ... If a track is a certain size at your wider layout, well it can be much smaller at your narrow one without you having to go and fiddle around with the item itself. It certainly makes it a lot easier to redefine at different break points and to redefine parts of your design based on different break points.

Jackie D'Elia: How do you think it might help with UX for design components? Is there any difference there?

Rachel Andrew: I think, I mean it certainly works very nicely with a pattern based and a component based approach, because you can look at each of these things, each thing in your design might be a small grid essentially. You've got different design elements that fit nicely into a little grid. I mean we've been using Fractal as a pattern library for our own stuff. I mean that's working with Flexbox at the moment mainly for the Perch UI.

We find that these modern techniques are much easier to work in this component based way, because you don't have to worry about the whole thing. You're not having to try and manage this grid structure which doesn't even exist, which is what you're doing when you're using float based layouts or whatever. I think there's lots of ways in which this can work quite nicely with that kind of component based, pattern library based approach which is becoming very popular.

Jackie D'Elia: One of the things I heard you mention on a recent podcast you were on, WP-Tonic, was just talking about how designers and developers can experiment with designs on their own sites specifically, and how things have changed over time with a lot of social media outlets and ways to publish content where we may not have an active playground that we use all the time for experimenting with design. I'm asking this because when I first watched a video about CSS Grid last year on a course I saw on lynda.com, I was thinking, okay well there's really no browser support for this yet.

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

It's not something that I need to have on my radar right at the moment, because it's not going to help me solve any of the current projects that I was working on. I really liked what you had talked about is having a playground to work that may not work in all scenarios, in that it's perfectly fine to build something out as a playground test site and start experimenting with new ways to design things. It seems like CSS Grid is a perfect opportunity to break the mold and try new things and experiment with layouts. I think it's really good to encourage people to do that.

Rachel Andrew: Yeah, absolutely. I think, at some point, the web got very grown up and professional and we're just doing our jobs. Perhaps we lost a little bit of the playfulness that we certainly had in the early days because it was all new. There wasn't really a right or a wrong way to do anything. We were just trying to do things and try and see what we could actually achieve with very, very limited technology. Now, we've got so many opportunities just within CSS. Even if you don't even think about JavaScript and all the other things that you could be doing, just within CSS, we've got so much opportunity to do interesting things. It's such a shame if everyone just replicates Bootstrap, you know?

I think on your own site, on your own project, you get the chance to do that. I've been encouraging people to look at things like look at magazine design. Say, "Well can I reproduce this?" It's not necessarily ideal for the web. It's not necessarily something you're going to put onto the next client project. It teaches you how to look at things in a different way and say, "Well can we actually build this with our new technology? Would we want to? How do we make it into a web design that has a bit of the feel of this magazine layout?" That's all quite interesting.

Jackie D'Elia: I think that's very helpful because it seems like most of the website designs are pretty basic in how we lay them out now. I mean typically you have a logo, you have a menu at the top. You have a hero image, you have some content blocks and a footer and that's pretty much the layout that you're going to go with. It doesn't leave a lot of room for creativity other than maybe doing some animations and, like you said, things in JavaScript that can change your layout and how it works. I don't think it goes far enough to really open it up and to start experimenting with new ways to do things on design. Some may work, some may not but it is really nice to encourage people to try new things and to have a place to do that.

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Rachel Andrew: Yeah, because we have the opportunity to do so. There's so much stuff that we could be using. It doesn't need to be in huge waves, you know? You can add little bits of delight and interest of designs using, say, something like CSS Shapes, for example. Wrapping text around things that aren't square. It's something we've wanted for a very long time and now we've got it, and very few people are using it. Part of that is because it's not supported in all browsers. You can use this stuff for little touches and let it fall back for the browsers. That's okay. Yeah, it's just playing with this stuff and just seeing what there is there.

Jackie D'Elia: That's a really good point about just embracing progressive enhancements and getting out of the mindset that it has to look the same in every browser.

Rachel Andrew: Yeah.

Jackie D'Elia: I think many of us struggle with okay, I need to find the common denominator that will work in all browsers for a design, and don't really want to add any other elements in that may not work in some browsers. Typically many of us have struggled with supporting Internet Explorer layouts and even in Edge, with some CSS animations and things like that. I think thinking about it in a different way that yeah, it doesn't have to look the same on every browser as long as it's usable and it meets the needs of the person visiting the site.

Rachel Andrew: Yeah. Look at the actual browsers that you're having to support and how many people are using them. Base this stuff around your own requirements and the sort of people that are coming to your site. Don't just assume oh well, this isn't going to work very well in IE9 or IE10. How many people are actually using that browser on your site? Could you be creating a far better experience for the 90% of people, and just allowing it to be slightly less designed for a smaller number. Every site's going to be different. Certainly on your own stuff, you really should be able to play and not worry too much if the design looks pretty plain perhaps in older browsers. You need to have these places because if you don't experiment, you don't learn how to use these things. You don't learn where the edges are.

Jackie D'Elia: The current browser support for CSS Grid right now ... And I think it was just February or March where the current versions of Chrome and Firefox and Safari supports CSS Grid?

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Rachel Andrew: That's right. It was March, yeah. All of those browsers got CSS Grid pretty much at the same time, within weeks of each other. That's never really happened before. This shows how much browser vendors care about interoperability these days. They want us to have all of the stuff in all of the browsers and be able to use it. Obviously each browser has a business behind it. They have things which they believe are more important than another thing that fits into the where they're going as a business.

Generally, they do not want to leave us with something that's broken in one browser over another particularly. They'd generally rather not implement something at all than give you something broken, because it's easier to check if something isn't supported than look for buggy support. Generally, there's a lot of effort to make browsers interoperable and to make so if you can do something in one browser, it's going to work in the same way in another. That's something people do care about.

Jackie D'Elia: How do standards evolve and then get adopted by browser manufacturers? How does that work?

Rachel Andrew: It happens in all different kinds of ways. Sometimes things come from a browser vendor themselves. For instance, Grid, it really started life in Microsoft. The first edits to that spec were from Microsoft. There's still one spec editor who is from Microsoft, and then there's someone who is an Invited Expert and then someone from Google and editors of that spec. A lot of stuff will come from a browser vendor themselves. Other people too, so the Shapes spec came from Adobe. Also then some stuff is just developed by the CSS Working Group because it solves problems, you know?

We can see that developers are trying to do a certain thing, and so there needs to be a CSS spec that does that thing. It'll actually just come out of the CSS Working Group itself. A lot of stuff actually starts life with either a browser vendor or there's an awful lot of people using CSS who are essentially in the print world. An awful lot of books these days are printed using CSS. There are very specific user agents that basically create print and PDF documents out of CSS and HTML. Things come from those areas as well but yeah.

Then it all ends up in the CSS Working Group and is worked on to try and make something that doesn't just solve the specific problem that maybe the person who first thought up this spec was trying to solve, but solves some general problems. You can see that evolution, if you look at the Grid spec as went into IE and what we've got now, you can see this evolution and how the set of problems it was trying to solve has been extended and improved and considering lots of different areas before it's been shipped.

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

- Jackie D'Elia: Okay. One question I did want to ask you was adaptive versus responsive design and the ability to reorder content now seems to be much easier. Even in Flexbox as well, you have that ability to reorder content. How can that help somebody create some really interesting adaptive designs that change based on the device that it's being viewed on?
- Rachel Andrew: Right. This ability to reorder stuff, we have that in Flexbox. We have it a lot in Grid. You can quite literally move anything around and put it into any area of the grid you want. You still really do need to take a great deal of care, because it's very, very easy if you start reordering content to make something inaccessible particularly for, say, someone who is keyboard navigating. If you're tabbing around, if you've moved an item that is in the DOM, at the top of the DOM and you move that down to the bottom, then what happens to their order as they try and move around the document? Both specs say that you shouldn't reorder the visual display over the source. You need to take great care if you start reordering at different break points, because you can quite easily make it very difficult to use the site which is not what you want to do.
- Jackie D'Elia: That would be containers that maybe have elements in there that require some interactivity?
- Rachel Andrew: Yes, yeah.
- Jackie D'Elia: With a user, right? If you just had a layout that just had images or just blog content to read, would that also affect your accessibility if you reordered that? If they were tabbing through?
- Rachel Andrew: Not necessarily. I think what you've got to do is take each of these cases as an individual thing and test it and say, "Well how does this behave?" I think there's so many conversations to be had around that. Essentially browsers will follow the source order. It's really, really important that you keep your source in a logical, sensible order rather than thinking, oh now I can use Grid just to move things around. Keep your source logical because that's going to be the order that the browser follows for anyone who is tabbing around or is using if that's being read out, for instance. Then that's going to follow the source and not the visual display. You just need to be careful not to disconnect the two completely if it matters what order things are in.
- Jackie D'Elia: Okay. What browser do you use for your development? What's your favorite browser?

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Rachel Andrew: I use Firefox a lot for Grid work, because they've got a Grid inspector and no other browser has. I often use Firefox. I use Chrome a fair bit, but I'm fairly agnostic to be honest. I tend to switch around which one I'm using. As I say, I'm using Firefox a lot just because they ship the Grid inspector and it's really good. If you're playing around with Grid, just the dev tools in Firefox has, if you select a grid container, so the thing you've said display grid on, and you open your dev tools, there's a little icon that looks like a little waffle. If you click the little waffle, you can see all your grid column tracks so you can actually see where the grid is on your page and on your element, which is really useful. That's definitely ... If you're using Grid, I would certainly be using Firefox just for the case of having those dev tools.

Jackie D'Elia: Any tools you can recommend for cross-browser testing when you're developing?

Rachel Andrew: I think really it's a case of getting your hands on as many browsers as you can. I mean that's ... So if you're on a Mac, I'm sure a lot of people are on Macs, the biggest thing is not having access then to Internet Explorer and Edge. You can get those in VMs from the Microsoft site. You can download virtual machines that have all different versions of IE and things installed. If you don't want to install a bunch of VMs, then BrowserStack is a good place. That's like an online site you can have a subscription there and log in and test things in different browsers.

For mobile, make sure you are actually testing on real devices, some real devices, not just emulators. How does this feel as much as what it looks like. I always say go out and get the cheapest, nastiest, smallest Android phone you can find, really low powered, and make sure you're also testing in that as well as the nice iPhone you've got on your desk or whatever. An inexpensive, the cheap phone that a lot of people have is a world away from an iPhone or the latest Android device.

Jackie D'Elia: Very good advice. The next topic I wanted to ask you about was Perch.

Rachel Andrew: Mm-hmm (affirmative).

Jackie D'Elia: I don't know very much about it and I'm curious. I heard you discuss it on the WP-Tonic podcast.

Rachel Andrew: Mm-hmm (affirmative).

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Jackie D'Elia: I know you had a great conversation about open-source versus non and sustainable business models. I'm just wondering if you can just give us some background on Perch and who is an ideal customer for Perch and how you operate it.

Rachel Andrew: Yeah. Perch is the product we launched. I mean we launched Perch about eight years ago as a small CMS so really, at the time, as a kind of content editor. The initial idea behind Perch was that you should be able to have a site that you've built in HTML and CSS, you didn't have a CMS, and then, of course, one day you're meeting with a client. The client says, "Oh you know what? I'd really just like to be able to edit the text on the homepage there." At the time, the only option you'd really have would be to rebuild your site into WordPress or something that would essentially mean you had to rebuild the whole thing, make a theme, take it away from being the site that was really just HTML pages.

The original version of Perch was like a drop-in CMS, the idea being you could just install Perch, stick a Perch tag on that area on the homepage where you wanted to edit ... Where the clients wanted to edit that little bit of content and that would be it. That was all you needed to do. You wouldn't have to change the whole site into a theme or really take over the whole thing with a CMS. People still use Perch very much in that way, you know? People build sites, static sites, and then just make bits of them editable. There's an awful lot more that we can do now, and we've got a second product called Perch Runway which is for your larger content based sites.

We care a lot about the idea that all of your templates should be completely controllable by you. Essentially if you're coming from a WordPress world, we're all custom fields. There's no concept of having a page that has to have a heading and a big area of text and so on. We're essentially all custom fields, so you define a template with exactly the fields you want. They load that up and that appears in the admin and that's what the client fills in with whatever kind of fields you want. That's really the difference is that we're very much based around the structured content idea.

Jackie D'Elia: How does open-source versus non-open-source ... What's your view on that? I know WordPress is open-source, so Perch is not, right? You have to pay for a license for each install.

Rachel Andrew: Mm-hmm (affirmative).

Jackie D'Elia: How does that work now for the clients that want to use ... What's their advantage to using Perch, say, versus WordPress?

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

Rachel Andrew: They have support included the license. They have support via us, the people who actually build it. If you have a Perch license, you can come and we will help you with your site which we do all day in the forum. You'll see both Drew and myself answering people's questions and help them out, so support's a big thing. Having a very opinionated piece of software is useful to people. One of the things with open-source projects is they can become quite ... Stretch off in various directions all over the place, because lots of people are contributing to them which can be great.

Also it can mean that things move in directions you weren't as keen on. People are very much, when they use Perch, buy into the way that we see the world, which is very much to do with the open web platform and standards and structured content and making the product very fast, you know? We will make decisions based on how fast that content's going to render over developer convenience, for example. We make those decisions all the time. They are things we care about. I think people buy into that. The support's a big thing. The support for end clients knowing that there is a company behind this, and a company that's been around for eight years and is making money doing this. It's not just going to disappear. That's very important to a lot of end customers.

Jackie D'Elia: I saw you can download Perch and try it out locally and experiment with it, which is a great way to see if it would suit a need for either you or for a client that you might have where maybe WordPress wouldn't be the best option for that.

Rachel Andrew: Yeah. I mean that's it. We have lots of people who use Perch alongside WordPress, who use Perch alongside Drupal. Most people aren't using a single solution because all projects are different. Unless you've really hung your hat on being a WordPress shop or we have people who are essentially Perch shops, they have hundreds of licenses because they just build Perch sites, but most people aren't doing that. Most people are getting a project and saying, "Well what's the best tool for this?" It may well be that you get some projects and you think, well actually Perch would be a great fit for that.

That doesn't mean that the next project's going to be, oh well actually WordPress is good for this one. We have a lot of conversations with customers based on the fact they know these other solutions too, and they'll say, "Oh. It seems like this feature would be really useful." We can have discussions about why maybe we don't do something that they like from something else. That's all really interesting too but yeah. Most people are using multiple solutions based on different kinds of projects that they work on.

Jackie D'Elia: I know I'm going to download it and play with it in my local development environment, because I'm very curious to see what it is and how it works.

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

- Rachel Andrew: Yeah. It's going to be interesting to hear what you think. As I say, we have a certain way of looking at the world based on where we've come from and the sort of things that we think are important. Obviously that matches with some people and not with others, and that's cool, you know?
- Jackie D'Elia: I definitely think the speed component is something everybody should be aware of and mindful of when ... A lot of themes in WordPress have an awful lot of bloat and create a lot of obstacles for getting the site to render quickly.
- Rachel Andrew: Yeah.
- Jackie D'Elia: That is something to be mindful of.
- Rachel Andrew: Yeah. I mean that's something that we've ... Because you have full control over your front end, we don't do anything to the front end. We make sure that actually rendering the content into your site is super fast, but then you can do any optimizations you want on your front end code because we're not going to touch it. That's very much how we work, because we don't have this idea of themes. It means people can really optimize their front end as much as they like. We always try and come up with ways. We were supporting things like responsive images very early and try and come up with ways that people can make their sites faster, because we care about it ourselves.
- Jackie D'Elia: Very good. We're almost out of time. I just wanted to circle back to CSS Grid and just talk about given the current browser support, which is just the very latest versions on there, how can a designer, a developer, right now start to incorporate CSS Grid into projects? How can they provide backwards compatibility?
- Rachel Andrew: Okay. Actually written into the spec is how Grid interacts with other layout methods. It interacts with them very well. What you can actually do is you can, say, have something which is laid out with floats, inline block, or just layout table and then also make it a grid. In an awful lot of cases, as soon as the thing becomes a grid item, all those other behaviors just stop applying. You can take advantage of the cascade and how CSS works to basically provide something for legacy browsers and then, immediately afterwards, provide something for things that support Grid, provide you the layout you really want, and that'll just work. The things that don't work, you need to look at feature queries which essentially is like Modernizr, if you've ever used it.

Transcript  
[rethink.fm](http://rethink.fm) Episode 14 with Rachel Andrew  
CSS Grid Layout

It's like using a media query except instead testing for the size of a screen, what you're doing is you're testing for support of a property. You can basically say `@support`, `display grid` and then inside that query, write your Grid code. Browsers that don't know about Grid, they'll ignore it and anything that goes along with it. Browsers that do will use it. CSS is bringing us ways to do this because it's always going to happen that there are going to be things not implemented in browsers. Yeah, on my site, I've got a cheat sheet of all those ways of overwriting older methods and links to stuff about feature queries and so on.

Jackie D'Elia: I guess the first order of business for most of us is going to be to start experimenting with this on a playground site or our own personal site, and then working with these cascading `@supports` and seeing how these things render. That sounds like that's a great way then to go and start incorporating some elements of that into client projects.

Rachel Andrew: Yeah.

Jackie D'Elia: It doesn't have to be an entire CSS Grid project. You can incorporate some of the CSS Grid pieces into a project that you're working on.

Rachel Andrew: I think what's going to happen is you're going to start, as you learn the spec and you learn how to use it, you're going to start seeing stuff in designs you're trying to implement and say, "That is a perfect case for CSS Grid." At that point, you can say, "Well can we use this to get the ideal layout, and then what can we do as a fallback?" I think increasingly people are going to start seeing things that can only really be done if you use Grid and doing it, and then working out what the rest of the browsers see. What's a perfectly acceptable layout, and then use Grid as the enhancement to get the real top of the line design. I think as people get used to having Grid, we're going to see more designers that take advantage of things that you can only do if you use Grid. We'll start seeing real use cases for that.

Jackie D'Elia: Then that wraps it up for us here. Thank you very much, Rachel, for joining me. If folks want to get a hold of you, visit your example website or contact you to follow up, how can they reach you?

Rachel Andrew: Actually I've got [gridbyexample.com](http://gridbyexample.com) is all my CSS Grid stuff and my personal site, which links to everything, is [rachelandrew.co.uk](http://rachelandrew.co.uk). I am `@rachelandrew` on Twitter and pretty much everywhere else. You'll find me, if you follow me on Twitter, you'll see all the stuff that I do.

Jackie D'Elia: Thank you very much again for joining me. I'll see y'all next time.

Transcript  
[rethink.fm](#) Episode 14 with Rachel Andrew  
CSS Grid Layout