DOWNLOAD PDF RESPONSIVE DESIGN ETHAN MARCOTTE

Chapter 1 : Ethan Marcotte on how responsive web design came about | Creative Bloq

Ethan Marcotte has just relaunched racedaydvl.com site and it is looking fabulously responsive as you might well expect. The attention to detail in this site is impressive, there are quite a few different media queries to adjust and restructure the layout of the page starting with a min-width:0 however there is a heavy use of calc in the font-size.

But as he sits in his white-walled office and sips his morning coffee, it seems the most logical place to start. First it was for myself, and then some side projects. From there I did some freelance projects for student organisations. I got my first studio job a year after school, and that was where the web got its hooks into me. They were people, he tells us, who had just stumbled into this new medium. It probably sounds a bit hackneyed, but I fell in love with this industry because of the people. Was the idea - that we should stop trying to tame the wildness of the web and roll with it - already germinating in various corners of the industry? Or did he have a classic eureka moment? The name, he tells us, did indeed come about suddenly and unexpectedly. The philosophy behind responsive, by contrast, crystallised more slowly. That this could be a place where people can do really brilliant things. I got really excited about moving away from this notion of fixed-width interfaces," Marcotte tells us. But in recent years, there was this school of thought - these responsive architects - trying to rethink that permanence. That clicked with me. I loved that image: And with that, the touch paper was lit. Responsive reaction So, how does he feel about introducing the concept to the web industry? Quite to the contrary, he warns against absolutism. And for me, that extends to responsive design. My goal is, irrespective of technique, to make the web as accessible as possible and as beautiful as possible for the people who are trying to access it. But I see responsive sites on a daily basis that are staggering and aesthetically beautiful. People are sending links [to the RWD Twitter account] that are beyond what I could have imagined when I wrote my article. People have taken this really simple formula of fluid grids, flexible images and media queries and are using that as a frame- work to design more flexibly for the web. Sub-Saharan Africa, for example, has a massive mobile market, far larger than what we have in the States and Europe. There has been a lot of criticism of sites that are two, three or four megabytes in size - just smaller than a PDF. So I think we need to have a discussion about sustainability in our work. Specifically, we talk about CSS features that Marcotte hopes to see ratified soon. As Marcotte finishes his coffee, we discuss the future.

Chapter 2 : Spotlight: Chris Ferdinandi â€” Responsive Web Design

Ethan Marcotte is an independent web designer who cares deeply about beautiful design, elegant code, and the intersection of the two. Over the years, his clientele has included New York Magazine, the Sundance Film Festival, The Boston Globe, and People Magazine.

You are the person who invented the concept of responsive design â€” or at least gave it a name, right? Yeah, I like to say I coined the term, because what I really did was I brought a couple different techniques, I think, under one umbrella â€” flexible grids , which is like the workhorse of responsive design. I like to think of responsive design as basically just like a rallying cry to bring us around to this idea of more flexible, more device-agnostic design. I think a lot of people have heard the term, and it gets thrown around and applied to a lot of things, but for people who might not be as comfortable with that concept, or who might be new to the term or not have heard it, how would you define responsive design for the typical web developer audience â€” web developer and designer, for that matter? I do a lot of consulting and workshops, some of them with Karen McGrane , a good friend and colleague of mine. So at the end of the day, in its barest form, a responsive layout is a web-based layout that uses flexible layouts â€” basically layouts that are defined in percentages or proportions rather than hard, inflexible pixels, and then they have images in media that are flexible themselves, that work in those flexible grids. Then, finally, the third ingredient is media queries , which are basically like a little bit of pixie dust from CSS3 that allows us to articulate when and how those layouts need to change and adapt based on different conditions â€” and more commonly, those are based on the size of the viewport. Do you want to talk about the origin story for you in responsive design? Yeah, I can give it a shot. There was a single aha moment, at the end, when I was working on this talk â€” this first talk where I coined the term. A big inspiration on my thoughts about web design was this article called A Dao of Web Design , which was written by a man named John Allsopp all the way back in So he was basically arguing that we should really try to treat the web as a completely unique medium, and design more flexibly for it. This being back in , when 4. So I would do a lot of flexible layouts for my own projects, for my blog back when I had updated mine , for side projects. But for most of my professional work, it was really very fixed-width layouts. Mobile then came along and broke everything. In the early days of mobile â€” at least in more developed markets â€” we were repeating the same process. I started getting more requests in client contracts for building an iPhone website, which was kind of weird. At some point, some other new cool thing is going to emerge on the mobile web. We could design these flexible layouts but still introduce a slight measure of control with media queries to articulate how to make these flexible layouts look beautiful on small and large screens. When you were looking at this, were you coming at this more from a design perspective, more from an engineering perspective? What background were you bringing to this? Not everybody can really come at a project from both a design and an engineering perspective. And I think there was confusion early on about do you call yourself a web designer, do you call yourself a web developer, is there even a difference? It may not necessarily be production-ready, but they could at least really effectively more clearly communicate their ideas to me. So, Ethan, one of the things that I think really contributed to the success of the responsive design movement is the ability of the industry to convince business leaders that responsive design was worth it. Do you want to speak a little bit to how that process folded out, and what, if anything, you learned throughout doing that? I think there were a lot of people contributing to that discussion â€” and still are, frankly. I met a publishing deadline, and I thought that was going to be the end of it. And folks just really ran with the idea. They started redesigning their websites, making them more responsive. I started seeing agencies going responsive, and then bringing the solution to their clients, and eventually I think that just trickled up the ladder. The same thing happened back in the mid s, when we were really starting to argue for CSS instead of table-driven layouts. You see this discussion happening at the fringes of the community, from independent designers and developers, and then it started getting some mainstream traction. I will say that I was involved with a much larger team of people on the responsive redesign of The Boston Globe back in , I think at that point we had a client â€” who was not a small publisher â€” who was super excited about this idea of going responsive, and how they saw it as a business imperative for them. When we took that live, they started talking about all the business benefits that they saw from that, and publishing those metrics and that data. And that made a lot of people in the media and publishing space sit up and take notice and that was really exciting to see. Responsive puts a lot more responsibility on the designers as well, and also on the people developing the content. Can you talk a little bit about that? I think, anything on top of that you can then look for those opportunities to really enhance the experience a little bit more. Virgin America is maybe my favorite example of this, where they basically said that they never showed their client at Virgin America â€" the design partner never showed their client at Virgin America a static comp. Basically, from their very first meeting, they had a responsive prototype â€" in a very early stage, but they had a prototype that everyone could load up on different browsers, on different devices. How do you recommend designers and developers work together on these things? Basically, The Boston Globe signed off on this beautiful array of comps that have been designed in InDesign. We basically need to continue the design process in the prototyping phase, and vet these assumptions in actual live web design, in an actual web-ready document. They were super receptive to that, which was surprising, and kind of exciting. Even the BBC who we had on the podcast last year talking about their global news redesign , they basically said the same thing â€" that thinking about UX and design as a separate practice from engineering and development, it actually slows down your velocity. They can look at Disney. I would say Twitter is maybe the primary way that I figure out new layout techniques, new design considerations, new questions and criticisms that are coming up around responsive design. And then I try to make sure that as many folks are aware of those other discussions as possible. I really like sharing that platform with as many people as possible. No, I mean, working on both of those books was actually a really rewarding experience, but I think any writing project is really difficult for me â€" especially in the middle of it. The second one that you asked about was about patterns â€" little reusable parts of design. The top-level grid thing I think a lot of folks are pretty comfortable with. But questions about, How do I manage fixed-width display advertising in a responsive design, or other kinds of third party content? Or What are some different approaches for managing response navigation systems? Because I think the principles thing has become such a bigger part of my practice lately. Is there another problem that you have your eyes on solving? This is the best and worst thing about this job, right? The web never stops changing! My name is Ethan Marcotte. You can find me on my website, ethanmarcotte. Otherwise, Twitter is probably the best way to contact me â€" beep on Twitter, or RWD for responsive web design â€" although I keep thinking of rear-wheel drive whenever I see that acronym. Those are the best places to find me. Beep is the best Twitter handle ever. I stand on the shoulder of GIF giants. Thank you so much for joining us and sharing all your wisdom with our listeners. It has been a real honor. Thank you so much to both of you for having me. It sounds like it struck him because it was something he needed in his own work, and he recognized that it was not only something he was going to need, but it was something that the rest of the web is going to benefit from as well. We can just be so grateful that he was willing to share that insight right from the very start. It really speaks to the strength of the community, and this industry. But you really need a large voice to get behind this idea and run with it. That, I think, is the most fascinating thing about the responsive movement to me â€" that it works to convince the people who essentially had the money and employed the teams to do all of this work. The Boston Globe was convinced to go ahead with responsive design, and it worked. It has such a broad implications for the work that people are doing. Yeah, the name was perfect. And speaking about the philosophy, it really is a hard shift to make, especially if you came into this industry making fixed-width designs. I mean, essentially, when you make a design in your responsive mental model, the design you make is your best-case scenario: It really is a tough thing to introduce to a host of people, both technical and non-technical. There was this philosophy about article writings. I was doing internal communications. We were producing a newsletter, and that newsletter was printed, on paper, and distributed to thousands of employees all around the world. Responsive design brings together that concept of the design has to adapt, the content has to adapt, to these different platforms. I really like how the idea of responsive design gave birth to other movements. I mean, if you think about it, responsive design was this renaissance period of thinking about the user. And then suddenly we started to say, Wait, how do you best want to view it, based off of what you have available to you? Then we started thinking about, All right, we got that part: But then we started to think about, How does that thing work for you? You see this very smooth transition from responsive design into the idea of performant web design, or designing for performance. I think progressive enhancement and responsive design, they are cousins in this context, because they really have so much to do with each other. All of these things are playing together. And for me, I guess, the concept of responsive design just brought it all together for people, and gave them something to focus on. You can even see how things like user experience and accessibility were strengthened by this idea. I think that I live in this bubble wherein I work on a product that is already responsive, and I continue to build responsive components. I spent a few years working at a company that specialized in taking big, static, browser-based websites and creating mobile-optimized versions of them that were dynamically generated on the fly from the desktop version â€” which was an amazing business model. And ultimately, is it just an API of content that then needs to be funneled into this presentation, versus this presentation for this audience, that audience, and it all gets customized? Yeah, one of the things that Ethan briefly touched on â€" he and Karen McGrane do a lot of responsive workshops, and one of the things that they always talk about is content strategy, because most of this does really come down to the content.

Chapter 3 : Ethan Marcotte | Examples | Responsive Web Design

I've spoken at conferences across the world, including Adobe MAX, An Event Apart, and Webstock. And I frequently offer workshops to my clients on a variety of topics, from pattern-driven design practices, to modern design/development workflows, to the fundamentals of responsive layout.

Everything you need to go responsive, in four short books. Why not bring it to your company? Subscribe Now Want the latest episodes? After years of struggling with hostile web forums, bad documentation, and incomplete tutorials, he now help beginners learn JavaScript faster and easier. He loves pirates, puppies, and Pixar movies, and lives near horse farms in rural Massachusetts. Hi, this is a Responsive Web Design Podcast, where we interview the people who make responsive designs happen. Because today I am pleased to be speaking with Chris Ferdinandi, who is here to talk to us a little bit about JavaScript that works everywhere. Chris, thanks so much for joining us. Thanks for having me, Ethan. So before we dive in, just to really quickly set the stage, earlier this year we did a series of spotlight episodes, where we interviewed designers, developers, and other front-end folks who are doing interesting work around responsive design, even though that might not be their primary focus. We thought that might be a good topic for the show. So Chris, welcome, seriously. I can definitely relate to that feeling. What was that path like? How did you actually start getting interested in front-end development? So, I actually had a human resources blog because my friends were so sick of hearing me rant about a lot of the stodgy old HR stuff that I thought everybody was doing wrong. So, I started writing about it, and I eventually hit a point where I was tired of working with whatever free WordPress themes I could find and I was too cheap to buy a really nice premium one. So, I wanted to start customizing it myself. It was one of those things where the more I did it, the more I found it more interesting than the HR work. In human resources, my focus was on teaching engineers how to do career-related stuff. So, I kind of took some of the stuff I was used to teaching those folks and applied it to my own career hunt. But yeah, it was one of those things where it was just a lot of hours of writing really terrible code and then writing it over and over and over again until it worked. Was that something that you had to deal with when you started getting your hands on front-end development? What kind of strategies do you put in place to deal with that? Back then, the most complicated thing you were really doing was maybe adding jQuery to your site before you started writing. But I often liken it to like skiing vs. Whereas with snowboarding, you spend the first three days just falling and you come out black and blue, and then once you get past the basics, you can start to kind of master your skills a lot more quickly. And that, to me, feels a lot like what JavaScript is like. So, from a basic level, that piece is kind of overwhelming. I also dealt a little bit with imposter syndrome. Once I kind of wrapped my head around that, I started to feel a lot better about the whole thing. So you mentioned earlier that a large part of your practice is working with other people and teaching them some skills for getting into JavaScript. So, the first is, hands down, the best way to get better at coding is to write code. This is not as difficult or as intimidating as I thought. Are you working with a framework? What kind of environment are you usually talking about? I personally am all in on just plain vanilla JavaScript. I personally learned how to code with jQuery first and then migrated over. For one, I feel like when you know kind of the root-level fundamental JavaScript stuff, you can go learn any framework you want a lot more easily than if you learn in one framework and then kind of have to mentally shift from the conventions of that framework into another. Knowing the conventions of different frameworks can be really useful, but without kind of that fundamental knowledge it becomes more difficult to pick up each one. So, React, for example, I think is awesome, but to really get the most out of it, you have to use pre-compilers and you often have to feed it through something like Babel. So, for all those reasons and more, I tend to shy away from frameworks whenever possible. So, talk to me a little bit more about fragility. How do you talk about that? Yeah, so it manifests in a few different ways. So, part of it is what you just referred to: In JavaScript, if you run into an error, everything dies. Even when JavaScript is appropriate, we use it in ways that make it way more fragile than it has to be. What kind of strategies or practices do you help clients put in place to deal with that? The first part is, for me, I have this philosophy of using as little JavaScript as possible, which seems counterintuitive given that my whole profession is centered around JavaScript. So, their page actually feels slower than if they just loaded it the traditional way. The other thing is not hiding content until the JavaScript loads. So, for example, you have one of those expand-and-collapse widgets. So, you take a drop-down menu, for example, and how you want to expose the content behind that. Traditionally, you might show content on hover. Sometimes it is; you have these really weird kind of hybrid devices now where you can touch and use a mouse and a keyboard and things like that. So, maybe you also want to detect click events or tap events. And then you get into some weird other behaviors, where if someone clicks on something or taps on something, a lot of browsers will fire a click event and a tap event. Those are, for me, those big things. Are they going to tap it? Are they going to click it? Are they going to hover over it? On which devices can they actually do those things? How do you get them to look beyond that device and actually put in some strategies that actually match up with their audience? I think either one is fine. Whatever you have more experience with. But one of the things that is often a concern for the people I work with, because in particular I work with a lot of freelancers who then go to work with clients, and so they need to have a lot of these conversations. So, a couple of years ago this would have been things like EventListeners and querySelector, which allows you to find elements on a page based on any valid CSS selector. The second part, for me, is I use terminal and I run some kind of prebuild processes with my code, but I absolutely hate how many articles and tutorials and best practices seem to center around these complicated tool chains, where you need to install Node Package Manager, or NPM, and then install a whole bunch of dependencies, and you end up with like a hundred megabytes of dependencies on your computer so that you can build a ten to twenty-kilobyte JavaScript file of stuff. It feels absurd and it adds a lot of complexity to the process. And so for that reason, I tend not to use tools that convert some of these more cutting edge JavaScript methods into stuff that older browsers can runâ€”and Babel is the one that most notably jumps to mind. Babel is kind of the tool that most people use if they want to convert the new modern stuff into backwards compatible stuff. But I like to provide support at least back to IE9, ideally even further back. With a polyfill, you can actually add support to some of those older browsers to do the same thing. It just makes the thing work in older browsers, which is great. Actually, just in the last couple of months, I switched from manually including my own to this absolutely amazing tool, called Polyfill. It was built by Jonathan Neal for Financial Times, and they open-sourced the project and made it available to anybody. Which, compared to name your favorite framework, even the latest version of jQuery, which is a lot smaller than the older ones, that is substantially smaller and allows you to just kind of write all the native stuff and work in older browsers, all the way back to IE7, which is really, really amazing. You mentioned file size and transfer size a couple of times. But the thing that I find most compelling about the whole thing is they have this really awesome screenshot view, where you can actually see, they take screenshots every one-hundred milliseconds, or a tenth of a second, where you can actually see how your page loaded in real time. And then when I can actually show them before and after examples, often which can be done by just re-ordering the way stuff loads or kind of changing a few things, suddenly they become really receptive to anything else you want to suggest around performance after that. I think the most important thing is that the content and the key activities that your users want to complete can be done in any browser or device, regardless of their capabilities. If someone wants to complete a purchase, they should be able to do that without JavaScript. If they want to view some content or jump around on a page, that should all work without JavaScript. If you really want to maximize your reach, you should explore using polyfills. Well Chris, this has been a really fantastic chat. So, seriously, thank you so much for your time. Thanks to everyone for listening to this episode of a responsive web design podcast. If your company wants to go responsive but you need help getting started, we offer a two-day onsite workshop to help you make it happen. We also offer these workshops to the public, so please go to responsivewebdesign. If you want even more from us, you can sign up for our newsletter, subscribe to this podcast, and read full transcripts of every podcast episode at responsivewebdesign.

Chapter 4 : About â€” Responsive Web Design

Ethan Marcotte is an independent designer and author, based in Boston, Massachusetts. He coined the term *"responsive web design" to describe a new way of designing for the ever-changing web, and is the author of the definitive book on the topic: Responsive Web Design.*

Chapter 5 : The Challenges of Responsive Web Design, with Ethan Marcotte â€” SitePoint

Responsive design's been around for a few years, but I still hear a lot of questions in my line of work."How many breakpoints does my responsive layout need?" "It feels like there are more devices and screens than ever before; how can my design process keep up?" "What exactly is the deal with CSS Grid?".

Chapter 6 : A Responsive Web Design Podcast by Karen McGrane & Ethan Marcotte on Apple Podcasts

Ethan Marcotte. Ethan coined the term "responsive web design" to describe a new way of designing for the ever-changing Web. His popular book on responsive design has been widely praised, as it demonstrates how designers and organizations can leverage the Web's flexibility to design across mobile, tablet, and desktopâ€”and whatever might come next.

Chapter 7 : Responsive Interview with Ethan Marcotte

By my estimation Ethan Marcotte's 'Responsive Web Design' looks set to become just as formative by calling for another new approach to designing websites - this time, one better suited to the ever-increasing number of devices and browsers that people us.

Chapter 8 : Responsive Web Design by Ethan Marcotte

Ethan Marcotte is a keynote speaker at Generate New York, which will take place on 20 June Enter the code generate at the checkout this week to get \$ off your ticket!

Chapter 9 : A Responsive Interview with Ethan Marcotte | Responsive Web Design

Ethan wrote the article that kicked this whole responsive malarky off back in and has been keeping us on our toes with new ideas and improved approaches. It is with great pleasure.. his answers to our standard four questions.