September 21, 2018 –

As told to Elliott Cost, 3645 words.

Tags: Technology, Design, Writing, Collaboration, Independence.

On exploring how to be online in radical ways

Web developer Tara Vancil discusses the peer-to-peer web, the current state of self-publishing, and the future of the internet. What was your first website like?

Oh boy. My first website is really embarrassing. I really didn't get into the web until about 2014. I didn't grow up online. We didn't have a computer at home, and even when I did finally get a computer at home, I wasn't using it for chat, or Geocities, or really any of the things that I think a lot of web people did in their teenage years. My first exposure to web development was actually in a programming class that I took in high school, and by programming, what I really mean is using Dreamweaver. So my very first website was a spoof on the *American Pie* films. I basically made a promotional website for a film called *American Cake*, and it was as heinous and gaudy as you can probably imagine it to be. It's stuck on a hard drive somewhere. It's in my apartment right now, but it's on a dead device, which is really sad because I remember exactly what it looks like. Oh goodness, how embarrassing.

That's amazing that you still have it.

Yeah. I mean, the markup is probably hideous because it was made with Dreamweaver, but I'm actually quite curious to see just how bad it is. But after that, when I genuinely got interested in building things on the web, came a personal website that I put together on GitHub pages. Building that website was a really painful process because I was in a mode where I was learning a lot for the sake of wanting to get a job building things on the web. I think that's a really bad creative place to be in, because I was looking at other people who I knew—looking at other people's websites who had jobs in web development—and thinking that I had to build something that resembled what they were doing.

So I spent a lot of my time browsing these websites and seeking inspiration, and trying to cobble together something that felt like me, and it never did end up feeling like me. At the end, what I put together was something really simple. It had a very basic navigation at the top. The main page was some blog posts I had written, and an "about" page. It's really funny that that process was so painful because now, building a personal website for me is like the most joyful experience. I'm actually currently redoing my personal website. It's a lot of fun because I don't feel constrained by seeking a job, and I have a much better understanding of how I want to build on the web.

I was reading your website and saw that you were part of the Recurse Center, prior to building Beaker. Can you talk a little bit about what that is, and what your experience was like there?

The Recurse Center is a special place. It's hard to describe because in a lot of ways it resembles a coding school, but it's not a coding school. It's a place for people who code who also want to spend time getting better at building stuff. So basically, they let programmers and developers come for a three-month period and just do whatever they want. There's no curriculum, no teachers. There are resources, and there are people around you who are interested in learning and growing. But that's it. It's a self-guided place to learn.

So I went in the fall of 2016 for three months. I packed up my bags, I moved from Texas to New York, and I spent three months in this building with about 50 other people, just programming. It was an awesome experience for me because it was the first time that I had some financial stability, and I had permission to just build stuff without worrying about looking for a job, and I desperately needed that. I desperately needed that space and that permission to just build, and I think that was one of the most transformative experiences I had as a developer. I highly recommend it to anyone who is new to programming, just starting off, or even if you've been doing this for 20 years. The people there are friendly, smart, and curious, and I think most people can have a really great time there. And it's free.

That sounds amazing.

How do you describe your project, Beaker Browser, to people who have never used it?

I like to say that Beaker is a browser for people who are optimistic about the Web as a space for creation, connection, and humanity. That doesn't tell you much about what Beaker actually is, so I usually

follow up by talking about how Beaker is different from other browsers because we're willing to run weird and outlandish experiments.

Our primary experiment is that we've put a peer-to-peer protocol in the browser, so you can visit p2p websites and apps, but you can also publish websites with Beaker. Beaker flips the idea that browsers are only for browsing, and suggests that browsers can (and should be) tools for creating and publishing, too.

I love that Beaker has a built-in editor. There's this all-in-one feel to it where you can browse and publish websites from the browser. I was curious what self-publishing means for you and why it's important?

Well, there's this myth floating around on the web that the very first web browser, it was called WorldWideWeb, made by Tim Berners-Lee actually had an editor built into it. Now, I've never been able to 100% confirm this with him, or anybody, but there's kind of just the shared history that goes around on the web, so I'm willing to believe it. When I found that out, it was really interesting because we had been building the early prototype of Beaker and it was quite different from what it is now. It did have a button that let you create a website from the browser, so self-publishing was a part of Beaker very early on. But we didn't fully understand how important facilitating self-publishing would be. It was fairly recently that we decided to put in an editor. We thought it would be too much work to maintain, we thought people wouldn't care, we thought they'd prefer to use their own editors. And then one day, we just realized like, "You know what? No, a browser really should help people participate in the web."
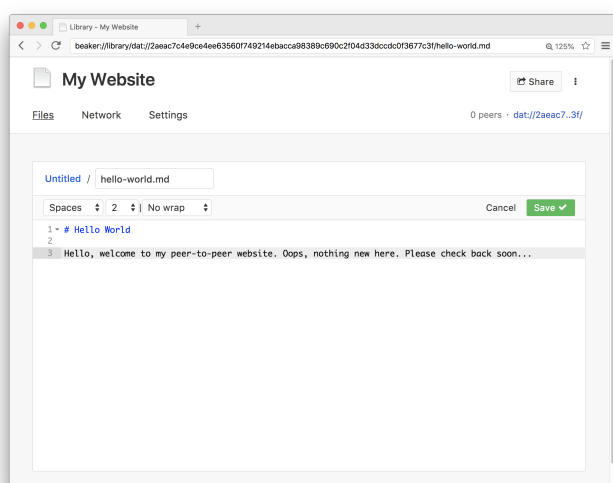
Beaker's in-browser web editor.

So self-publishing, for me, is not necessarily about owning your content. It's not all about enabling creativity. There are other tools that enable creativity. I think it's about creating opportunities for the widest swath of people to participate on the web. I think right now, there are so many barriers that can pop up at any given moment when you decide, "I want to make an app, I want to make a game, I want to publish my portfolio, or I want to create an interactive art piece." With Beaker, self-publishing is about reducing as many of those barriers as possible, so that literally everybody can have some hope of meaningfully participating on the web. Because why not? That's what the web is. It's this really strange thing.

I like to call the web humanity's shared language. We've all come together, by some miracle, as a society to define a set of rules and technical standards about how we will communicate, how our computers will communicate with each other, and people all over the world use this. I mean, that's pretty miraculous that we've managed to do that. So why shouldn't everybody be able to build stuff on it, and share things on it? It seems really sad that right now that's not the case, and I think it's also boring.

There seems to be a general feeling that HTTP doesn't provide a productive space any longer. Recently there's been a lot of interest in going offline or just slowing down. I wanted to get your thoughts on the offline first movement and if you align yourself with it?

Offline first is a funny concept to me because it's rooted in both very corporate ideology and very anti-corporate ideology. So there's one meaning for offline first, I think it was coined by Google, and this was a way for building applications such that low-power devices in places that have really bad connectivity could cache an application's or website's assets so that it can still function well. I think this is an honorable effort to build applications with the expectation that we don't live in an equitable world, but we have to remember that a corporation like Google is motivated to do that because they want to

sell more devices, and they want to further the reach of Gmail and their other tools.

And then there's the other side of the movement, where offline first means something very different to another group of people. If you've heard of Secure Scuttlebutt, it's a peer-to-peer online friends space. It's a place for people to post content and share things with their friends without having to connect through something like Twitter or Facebook. And a lot of the folks that participated there in the early days were really interested in finding ways to live a little more independently, to maybe not depend entirely on the electrical grid, or to be able to live on a boat, or to maintain their own garden. I think that reflects an interest in slowing down, and a reaction to the speed of consumption that the web of today demands of us.

So at the end of the day, I think offline first—by both definitions—is rooted in the observation that we don't live in an equitable world, and modern applications do not serve everybody. They don't serve every kind of lifestyle. I'm definitely interested in living in a home with electricity and modern amenities but I'm also really interested in doing that responsibly, and I care a lot about my own sanity and other people being able to maintain their sanity in this hyper-connected world. I think a lot of us are perhaps exploring how we do that for the long-term. So I like being online and I want to continue being online, but I think looking to these communities that are exploring how to be online in radical ways, is really important.

Beaker is a good example of that. In my own exploration of the peer-to-peer web, I've needed to either be sent a link directly from somebody, or be in connection with the HTTP web to find websites on the p2p web. I'm curious what the longer-term goals are? Is it sort of like in tandem with the current web, or is the goal to replace HTTP with peer-to-peer protocols?

Yeah, there's an interesting effect on the peer-to-peer web where you kind of have to bootstrap your experience somehow. You either have to have a chat open with a friend so that you can send links between each other, or you need to have a curated list of websites and projects that you want to visit. And interestingly, I think that's a problem that the HTTP web suffers from as well. It's an aggregation problem. If you think back to the early days of the HTTP web, someone—or some company—had to go out there and crawl the web, and collect the links that they found, and then publish them somewhere. That's just a fact of how networks work. It's hard to aggregate content independently.

So I think what that means is that if the peer-to-peer web is going to become a part of the web as we know it, then so are search engines and aggregators. And maybe those search engines will use HTTP just because it's easier for that purpose. Maybe not. I'm not sure that we need to replace HTTP entirely to fix what's wrong with the web. I think we need to replace HTTP in cases where it encourages centralization of governance over our communities, and it discourages innovation and the ownership of our online experiences. That's why I think it's so important that people are able to publish their own websites, for example, because a website can be anything. It can be the place where you post your micro-blogs, like your tweets. It can be a place where you post blog posts, which is pretty obvious. It could be a place where you post photos or art projects, and I feel that the HTTP web makes it so difficult to do that right now. As a result, we're cornered into the situation where we have to publish on Twitter, and Facebook, and Instagram. And that's fine, those are pretty cool platforms, but they also constrain us, and I think we're starting to understand the limits and the consequences of that.

It might be a positive thing that you can't search the peer-to-peer web currently, in that it has to be such a personal connection where my friend will send me a link to a website. HTTP is a constant process of following links to other links. On the p2p web it's more about accessing a page and then reading it to the end, and then maybe going offline after that.

Yeah, there's a certain finiteness to it, which is blissful at times. I'm not sure it'll stay that way forever. There's a lesson to be learned about how it feels to use the peer-to-peer web. I've found websites where I couldn't believe I found them. It felt like I'd just stumbled upon a treasure. Like, "Wow, this person is out there and they've made this thing. I want to read everything they've posted," and then that's the end of it. It's a really satisfying experience.

It also feels like you have to forget what you thought the web was when you're approaching the p2p web. I find it pretty difficult to describe what the peer-to-peer web is, and I think maybe that's not just me. It's broad, it's many different things, it's multi-layered.

What does your ideal web look like?

I want a web that I can build on. I love building on the web so much. To me, websites are my canvas. I grew up in a family that I think looked down on anything that smelled of creativity. I grew up hunting, watching football, and playing sports. There's no creative exploration in that. I became exposed to the creative process fairly late in my life, and the canvas for me is websites. I love the feeling that I get when I sit down with a blank slate, and I know how to use the tools, I know how to wield HTML and CSS and bend it to my will. I want a web that is conducive to that, and I don't want to just build standalone websites. I would love to build things that are meaningful to people, that have users, and then I want those users to be able to take what I've made and be able to shape it into something new.
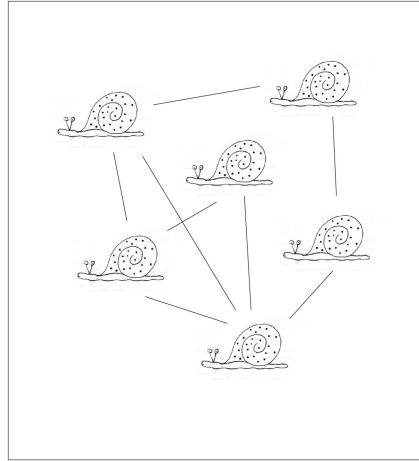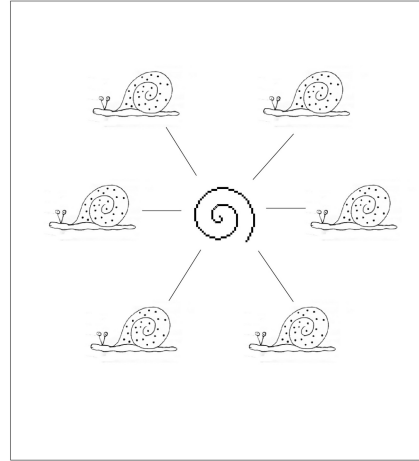
decentralized (p2p)   centralized

On the web today, I feel like I can build something amazing, and I can go out and find people who want to use what I've built. But it's a very rigid process. To build something, I first of all probably have to find investment because launching a service on the web, launching an app that's actually going to get wide usage, is really, really expensive. So I think I want a web that makes that process cheaper, and distributes the cost of bandwidth and storage across its users. And then beyond that, I want a web that doesn't try to lock down the experience of using any application because there's no way that I can build a perfect app that suits everybody's needs. Everybody has different needs, everybody understands what their needs are as they use an app, and those needs are probably going to evolve.

I think it's a shame that the web today isn't conducive to change and exploration, and I think that is HTTP's fault actually, because it is so expensive to run a server, and servers are kind of like the boss. If I build an application and I launch it on a server, that app, the code for that application, it lives on the server and the server dictates how that code works. A user doesn't have any access to that code and they have no influence to convince me that the code should work differently. I'm really hopeful that we can build a web that gives agency to more than just servers.

In Beaker, there's a feature where you are able to seed websites for a certain amount of time. It's just a slider that goes from one day to forever, which is really funny. I feel like it's difficult to build software that has that sense of time and longevity built into it. I also just love the idea that you can help somebody keep their website online forever.

Oh gosh. So the concept of seeding or peering or re-hosting stuff on the peer-to-peer web and in Beaker has been a really hairy and challenging problem since the beginning. I think, first of all, because it's a brand-new concept to web users. I mean, if you've used BitTorrent or any other kind of peer-to-peer protocol before, you might be familiar with the concept of seeding. But I think most people aren't, so we had to figure out, "Okay, we know this is a really important concept on the peer-to-peer web because it's how we reduce the cost of publishing. So we have to communicate it, we have to share it with the people who use Beaker. But how?" And we went back-and-forth for a long time on which words to use. We didn't really like any of them better than the others. We kind of just said, "You know what, seeding has a nice tone to it. It reminds me of a plant, a seed growing, so let's just go with it because it sounds nice."
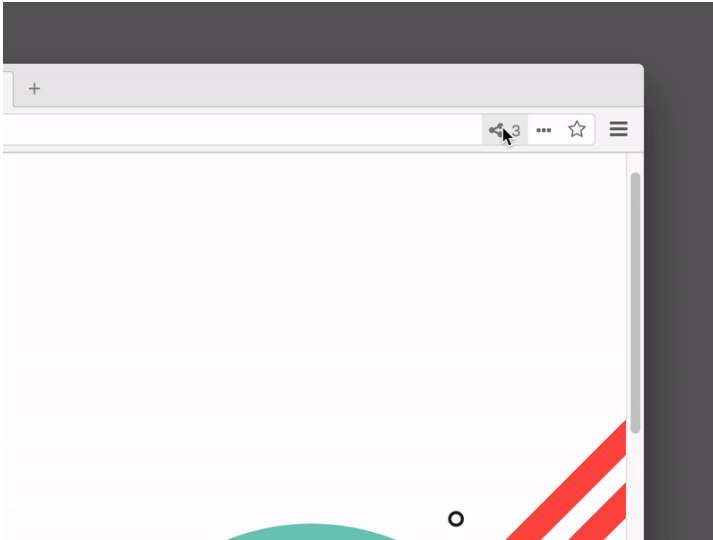
Our first inclination for building the interface for seeding a website was, "Let's just do a toggle. On or off." But that didn't feel right. It felt like it didn't tell the person anything about what it means to seed something. We realized that the temporal nature of seeding is really important because it ties in nicely to the concept of virality, something going viral on the web. There might be an important news story published, let's say, and it's situational. It depends on the moment that it was published in, and you want to make sure that you're helping publish it during that really heated moment when it's spreading quickly. But you don't want to seed that forever, and you're probably not going to remember to turn off the seeding toggle. That realization that time matters because the web is not static was important. The web is changing, and the things that we care about on the web change over time. That was a really big realization for us, and we've been pretty pleased with how that UI element has worked out since.

This interview includes some technical terms:

HTTP: Hypertext Transfer Protocol. This is currently the most widely adopted web protocol in use today. Every major browser, from Google Chrome to Mozilla Firefox, uses HTTP. HTTP has been around since the beginnings of the internet.

Beaker: A new web browser built by Tara Vancil and Paul Frazee. Beaker allows users to connect to the peer-to-peer web via Dat (dat://). More on Beaker can be found here.

Peer-to-peer (sometimes shortened to p2p): Peer-to-peer has been around since the beginnings of the internet. On a p2p network, users help host each other's files. This is called seeding. When a user requests a file, all the computers in the network share the bandwidth needed to complete the download. BitTorrent and Napster are examples of p2p networks.

Dat: Dat is the p2p internet protocol that Beaker is built on top of. Dat can be thought of as a Dropbox with peer-to-peer capabilities. If you think about it, a website is just a collection of files, so Beaker is just a way to view those shared folders in a browser. More about Dat can be found here.

Decentralized web: A general idea to describe a web that is not centralized or stored on a central server such as a server farm. Peer-to-peer is an example of a decentralized network. The network is spread across many computers that are controlled by individual users. This redistributes control back to users.

Recommendations

Information Civics by Paul Frazee. Paul is the original creator of Beaker and my chief collaborator. He also used to be a developer on Secure Scuttlebutt, so he has a lot of practical experience exploring governance and power structures on the Web. This project is his effort to put that exploration to words and ask "Why decentralize?", then provide justification for all the trouble decentralized networks involve. Also available at dat://infocivics.com

Spifster Sutton's nail designs, especially this one

EVERYTHING IS LOVE, The Carters's (Beyoncé and Jay-Z) debut studio album

Water. Drinking it. Floating, splashing, submerging yourself in it. Listening to it. Keeping your cats away from it. But maybe also flicking their backs with tiny droplets because it's adorable how confused

and annoyed it makes them.

Name
Tara Vancil

Vocation
Web developer

Fact

Related to Web developer Tara Vancil on exploring how to be online in radical ways:

The Creative Independent is ad-free and published by Kickstarter, PBC. See also: Terms, Privacy Policy.

↑