

If you're in the position of needing to interview someone about their skill and knowledge about CSS, it can be a little hard to think of things to ask on-the-fly. I thought I'd think up and round up some ideas for reference.



Exercises To Do

Seeing people's actual work is just as important as what they say. Seeing people work through exercises live might be even more important. These are some exercises that aren't particularly difficult and anybody with CSS experience should be able to do. Watching/listening to them do it could be invaluable. Collab Mode on CodePen is kinda perfect for that (just saying).

Create This Button

I saw this idea on Mobify's CodePen account. Give people an image of a button, and tell them:

Using CSS properties alone, recreate this button:



This would be a great test of the candidate's CSS3 skills. This button is loaded with that kind of stuff. Things I would look for:

- How did they handle multiple borders? `box-shadow` is probably the best way here.
- Did they use `text-transform: uppercase;` for the text? That would be best.
- How did they handle the stars? Pseudo elements would be a good candidate there. Did they use unicode? Icon font?
- Did they catch the subtle `text-shadow` ?
- How did they split the background in the main part of the button? a `linear-gradient` with no fade would be a good technique there. Or `box-shadow` .
- Were they careful with the nested border-radius?

I hope I didn't ruin Mobify's hiring process! But honestly, I think this stuff is hard to fake. You can either do it and talk about it, or you can't.

Fix The Sidebar

The right sidebar here has fallen down below the content. Show me some different ways you could fix that.



Even though the two columns are 75% and 25% wide and floated opposite ways, the sidebar has fallen. The reason is because the columns don't actually add up to 100% - they add up to more than that due to the padding. There is a number of ways to fix it:

- Using `box-sizing: border-box;` on the appropriate elements is the most efficient fix.
- Using `calc()` on the widths to remove the 1rem padding is another way.
- Putting the padding on an added internal wrapper element instead of the columns is a fix that will work with very deep browser support.
- Adjusting the numbers to make the math work is another way. For instance making the widths of the columns 4% narrower and using 2% for the padding instead.

There are other ways. The more solutions they can think of, the more creative and versatile a problem solver they are.

Make this fixed width design fluid.

Here's a design. It's a fixed 800px wide. Do your best at making it fit the screen more appropriately at any screen

size.

This is just one ingredient to responsive design (which they should probably at least be familiar with) but it's an important one. It can help prove they can think spatially and make reasoned choices about layout. I'd look for:

- Changing the pixel widths to percentages (how did they handle the math?)
- Did they do anything special for large screens or just small?
- Did they attempt to use a responsive images solution?
- Does the new design retain the hierarchy of importance inherit to the original?
- Did they come back at you with questions? (Lots of things to ask here, including what other resources might be available.)
- Did they test it? (To make sure it actually works, and find things like missing meta tags.)

Replace this logo markup with an image.

HTML

```
<a href="/" class="logo">Company</a>
```

Accessible and semantic image replacement has been a CSS topic for a lot of years and the "best way" has morphed over the years. Asking them to show you how it could be done a number of ways would be a way to get insight into how long they've really been working with CSS. Not only is it directly an important thing to know how to do, knowing how to do it multiple ways demonstrates the depth of their internal toolbox.

Google how you would find out what the default value for backface-visibility is.

Being able to Google something quickly and efficiently is a huge part of any developer's job. Are they deft at it? Did they find the right answer? Did they go to a specific trusted source in the results page?

Perhaps phrase the question without "Google" and see which search engine they use. No particular bias here, but if it's not Google, are they as efficient as you would expect them to be with whatever other search engine that is?

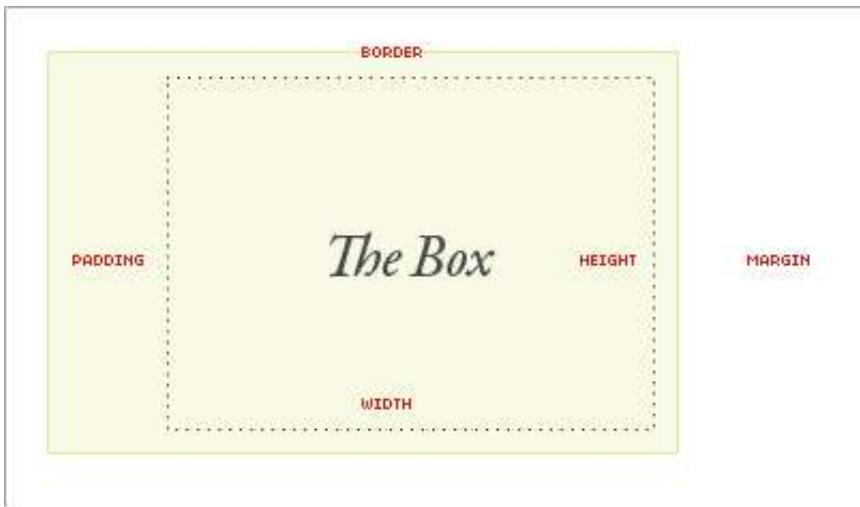
Questions to Ask

Darcy Clarke got the ball rolling with this a few years ago. I'm going to update some of those, and add some of my own, and elaborate a bit.

What is the "Box Model" in CSS? Which CSS properties are a part of it?

The CSS box model is fundamental to understanding layout and sizing and such. It's made up of:

- Width and height (or in the absence of that, default values and the content inside)
- Padding
- Border



Margin is related but not technically a part of it. I would give extra points for knowing/mentioning that, but not take away and points for including it.

What are Sass, LESS, and Stylus? Why do people use them? How does something like Compass relate to Sass?

They are CSS preprocessors. They are an abstraction layer on top of CSS. They are a special syntax/language that compile down into CSS. They make managing CSS easier, with things like variables and mixins to handle vendor prefixes (among other things). They make doing best practices easier, like concatenating and compressing CSS.

Bonus points for knowing how they differ and/or having experience using them. More bonus points for knowing what things like Compass, Bourbon, LESSHat, Nib, etc are and how they relate.

Name some online resources that you reference when having CSS issues.

Being good at googling problems you are having is a valuable job skill. There isn't any shame in it. There is shame in spinning your wheels because "you should know this." If you don't have time to do the Googling exercise above, just asking about resources can be telling.

Google is a pretty good answer (since it's true and we all know it). But being able to name some specific sites is a good indicator they have done it a bunch and are familiar with the places they land and know their favorites. Stuff like MDN (Mozilla Developer Network) is a good answer.

Describe what a “reset” CSS file does and how it’s useful. Are you familiar with normalize.css? Do you understand how they differ?

Resets are so wildly common in CSS that anyone who is a front end developer type has surely used them. Do they do so blindly or do they know why? The reason is essentially that different browsers have different default styling for elements, and if you don't deal with that at all, you risk designers looking unnecessarily different in different browsers and possibly more dramatic breaking problems.

Normalize you might call a CSS reset alternative. Instead of wiping out all styles, it delivers a set of reasonable defaults. It doesn't unset things that are already consistent across browsers and reasonable (e.g. bold headers). In that way it does some less than a reset. It also does some more than a reset in that it handles quirks you may never consider, like HTML5 audio element inconsistencies or line-height inconsistencies when you use sub and sup elements.

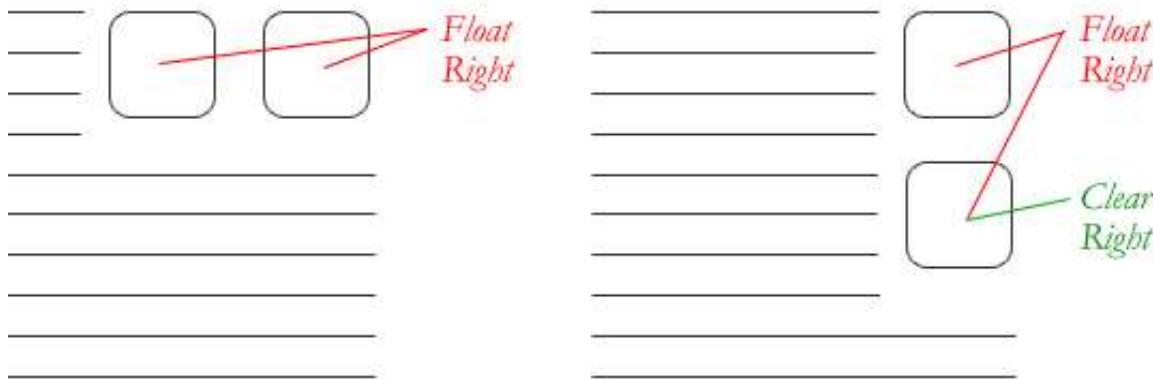
What are the various techniques for clearing floats?

Floats are still incredibly common. As this is published, still probably the most cross-browser consistent way to build layout and grids. Anyone who has worked with them is aware of float collapsing. That is, floated element do

not add to the height of a parent element. So for example if a parent element contained only floated elements, it would collapse to zero height. You can deal with that like:

- Use a clearfix (bonus points for micro clearfix).
- Float the parent as well.
- Use an overflow property other than "visible" on the parent (bonus points for listing downsides like cutting off shadows).

Bonus points for "create a new block formatting context". Possibly negative points for something like `<br style="clear: both;">`



Do they understand directional clearing?

As a bonus question, you could ask them to compare using inline-block and floats for building a grid. Good answer: there are problems either way. With inline block you need to deal with the whitespace issue. With floats you need to deal with clearing.

What are sprites and why would use them? How do you go about creating them? What are possible alternatives to sprites?

Sprites are essentially multiple images combined into one. Performance is the reason that they are used. Generally speaking, the slowest thing a website can do is request a resource. The fewer requests a site needs to make, the faster it is. Fast = good. Combining what would be many requests into one = good.



Asking how they make sprites would just seal the deal that they are actually very familiar with them. Manually creating sprites is certainly a possibility but it isn't very efficient. There are helper tools like SpriteCow and SpriteMe, Spriting with Compass, or Grunticon. It's always interesting to hear a real workflow approach.

Sprites are raster images. When asking about alternatives, good answers might be related to the fact that sprites are often for icons and icons often don't need to be raster. SVG Stacks, Icon Fonts, Unicode...

What are some accessibility concerns that come up in CSS?

Hidden content is a big one here. It's only acceptable to use `display: none;` if you're both trying to hide things

visually and the content itself.

CSS controls colors so color accessibility is relevant. Focus styles are also important and directly controlled by CSS.

There is a lot more accessibility stuff that is related to HTML and JavaScript, so mentioning that stuff is great, but I think it's interesting to target the question solely on CSS to force focused thinking.

What is the difference between inline, inline-block, and block?

Bonus points for bringing up specific details like the fact that you can't transform inline elements.

What tools do you use for cross-browser testing?

They should have some kind of strategy. Perhaps a web based tool like BrowserStack. Perhaps a VM based tool like Virtual Box. Perhaps different actual computers.

Part of the job of front end design is making sure things work everywhere they can (based on decided-upon support). You don't have to love it, but you can't hate it. *"This right here, this is the job. What kind of work were you expecting?"*

What are some of your favorite web design workflow tools?

What code editor do they like? Where do they go for inspiration? What experience with version control do they have? What was QA like where they have worked? Support? What different deployment methods have they worked with? Do they know Photoshop or another visual design software alternative? Are they comfortable with the terminal?

Those are just some examples, it's interesting to hear about any software they use to get the jobs done. Getting a sense of the tools they use (and better, the tools they *like*) is interesting. Bonus points: a sense of excitement about some of the tools.

Say you found a rendering problem on one of your sites in Internet Explorer 8, which you have decided you are supporting. How would you approach fixing it?

This would be an alternative question to asking generally about cross browser testing. More specific. Perhaps a more difficult alternative question would be to replace IE 8 with something like "A Google Nexus running Android 2.3." Would they find a simulator? Would they seek out a design lab? Would they ask the company for an allowance for devices for testing, with some kind of smart plan? Would they find a friend with one?

What is responsive design all about?

It's about making websites work wherever the web is. Different devices with different sizes and different capabilities. Responsive design is about taking one code base and making it work for all of them. Part of that is media queries and different visuals. Part of that is different resources (e.g. different JavaScript to handle touch vs click or different images to accommodate the screen).

Have you ever worked with a grid layout? What are your thoughts on that?

Why did they need a grid? How did they build the grid? Was it home grown or did they use a grid tool? Did they like the grid tool? What kind of class names did they use? Did they go mobile-first or desktop-first? Was a help or a hinderance? Do they reach for the grid automatically on any project?

What are the benefits of SVG?

SVG is an image format that is vector based. It's an efficient format for that (small file sizes). You can scale them and they retain their sharpness at any size (bonus points for mentioning raster might have the upper hand at tiny sizes). You can affect parts of them with CSS and JavaScript as well as SVG specific filters that can do things like blurring.

Have you ever created a print stylesheet for a website?

Kind of indicative that they've "gone the extra mile" with websites before. What approach did they take? How did they test it?

Say you were tasked with coding a design that used non-standard web fonts, how would you go about it?

A non-leading way to get them to talk about @font-face and how it works. Talking about how it works as a core CSS technology is great, as well as talking about services that provide the fonts and can make it easier e.g. Google Fonts, Typekit, Font Deck, Cloud Typography, etc.

Bonus points for obscure knowledge like the history of @font-face syntax or Firefox's issue with cross-origin fonts.

Explain to me what's going on in this CSS selector:

CSS

```
[role=navigation] > ul a:not([href^=mailto]) {  
}
```

This selects anchor links that are not email links that are decedents of an unordered list that is the direct child of any element with a role attribute of 'navigation'.

Being able to verbalize a selector is proof you understand them and evidence they can communicate complex tech subjects.

Your Ideas

What kind of questions have you been asked when interviewing for a CSS-related job? Or that you ask your candidates? Or that you think would be good questions in general?

SHARE ON

[Twitter](#) [Facebook](#) [Google+](#)

Comments



Jared Christensen

OCTOBER 14, 2013

Have you seen this... <https://github.com/darcyclarke/Front-end-Developer-Interview-Questions>



Chris Coyier

OCTOBER 14, 2013

I linked to that at the top of the **Questions to Ask** section. Some of these questions are based on those.



Scott

OCTOBER 14, 2013

“Whiteboard” style coding-on-the-fly exercises are so far beyond useless and a waste of time that it makes me angry every time someone suggests using them. You’ve got a stranger sitting in a room with a couple of other strangers and handed a piece of paper or a whiteboard marker and told to create something?

Expecting someone to write anything that isn’t garbage in an environment *about as diametrically opposed to producing good code as is possible* is insulting to the interviewee and far less an effective use of your hour (or so) than looking at their previous work and discussing it.

My \$0.02.



Chris Coyier

OCTOBER 14, 2013

I pretty much agree with that. Although I don’t think asking them to code something at all is out of the question. Maybe like, here you have 20 minutes to work on this. Work on it (perhaps even at home alone) and then we’ll talk about it when you’re done. And the subject is also important. Whiteboarding a sorting algorithm from the 60’s is ridiculous. Asking someone to CSS a button is mega practical and evidence they’ll either be able to do the job or not.



Melindrea

OCTOBER 14, 2013

I think that I will echo Scott's comment because it can't be said enough times!

Testing possible candidates on coding == good. However, those tests should be handled through "homework", a work test to be sent in before or after an interview, not at the interview.

As you say, Chris, seeing how they work through things and that is a good thing. For one, I got the job I have because of ordered and good quality code, where others weren't interested in the task because it was too daunting.



Laurel

OCTOBER 14, 2013

Yes, if I was asking someone to actually create something complex. I like the button example here, because anyone who's fluent should be able to rough it out. The properties don't have to be perfect—for example, I wouldn't mark off for getting the order of the box-shadows reversed—but they should be able to draw out the structure and explain what properties create which effect.

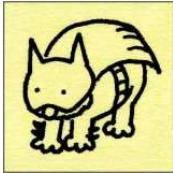


Don

OCTOBER 14, 2013

I would strongly disagree with the scope of your position. Asking someone to write something that's syntactically correct is insulting and a waste of time. Asking someone to produce a basic description of a way to attack a problem and explain their reasoning isn't, particularly if you're looking for someone experienced enough in the field to be versatile and lead others.

If you're going to ask that they produce anything beyond a simple solution, particularly to a debugging problem, yeah, this is a bad way to do it. Expecting them to remember everywhere something is two conjoined words vs two hyphenated words is unreasonable. You may well be hiring for a position where you can just throw raw meat down a shaft and have them throw designs up; if that's the case, or you're looking for someone sufficiently junior, fine. But the interview where you expect someone to show you how they arrive at a solution and how they'd explain things to others wasn't antiquated by Google and just-in-time syntax checking. If anything it made it more important for certain levels & needed competencies.



Peter Hentges

OCTOBER 14, 2013

I'm going to agree with Scott here. Being asked to do these kind of exercises in an interview would be a big red flag for me joining the company. It strikes me as indicative of an environment in which things are far too tightly wound. If someone's insisting on doing these sorts of test, making them take-home assignments to be completed and reviewed would be a better approach. Better yet would be looking at examples of work and seeing the style of work the person generally displays.

That said, some of the broader questions like “what tools do you use for x” or “what do you think about grid layouts” are things that would provoke some interesting discussion and show that the interviewer was familiar with some of the technical details as well. I’d welcome those in an interview.



Simon

OCTOBER 15, 2013

I mostly agree with Scott.

That said if hiring for a seniorish position, I think it is fair for candidates to bring their laptop along and do a little bit of coding. That shouldn’t be a red flag, but a chance to prove your chops.

I’d always advise candidates well in advance that this part of the process and do it at second interview stage, maybe by spending some time working collaboratively on a genuine problem.

I don’t really care if you know syntax backwards, but I do want to see “live” evidence of depth of knowledge (appropriate to the position), code organisation and a creative approach to solving problems.

This should be the part of the interview process we all enjoy most.

My \$0.02 from the hirer’s perspective.



Nojan

OCTOBER 15, 2013

Couldn’t agree more with scott. these kind of questions mostly demonstrate the lack of knowledge of interviewer (by the way chris i’m not referencing you).

If it wasn’t so, he/she could analyze the applier’s recent works before meeting with him and then ask him more general questions like work style, etc..

As for myself, i don’t have a very good memory. So if someone asks me about different ways of replacing image, i may not be able to answer him at the moment. but i have enough experience to figure it out the best way possible each time and i think that’s what is important.

But anyway this article is very informative ;)



Scott

OCTOBER 15, 2013

Sorry Chris, I didn’t mean to take over the conversation with my little rant, and hopefully the resultant discussion didn’t take away from what is really an excellent article and series of questions. The “Questions to ask” section covers a very good, broad range of skills and knowledge that would give an interviewer a strong sense of a candidates’ ability.



Ricardo Zea

OCTOBER 15, 2013

@Simon, I don't use a laptop to do my work and honestly don't plan on getting one any time soon (I've built my own rigs for years). So maybe "bringing your laptop" isn't the best approach either.

Employer should provide the laptop/desktop PC or any 'testing' tools regardless.

Now, working together for 15-20 minutes to solve a current real life problem in the company... agree.



Andrew

NOVEMBER 29, 2013

I whole-heartedly agree. I've been in several interviews where, for a job that clearly would only have me sitting writing CSS and some basic HTML, I was required to come up with a regex that encapsulated US phone numbers or emails from certain sites. Sure, this would make some sense if I am sitting writing tons upon tons of JS but as someone in a mainly design-oriented position it is just asinine to ask of. Even asking of "psuedo-code" for a layout is terrible. Is it to be responsive? Where in the browser-compatibility scale are we shooting for?

As Chris says, a static challenge, even with a more flexible timeframe is much more reasonable and at the very least marginally productive.



Senff

OCTOBER 14, 2013

Not directly technology related, my favorite question whenever I interview someone is how they rate themselves as a front end developer on a scale of 1 to 10. It's usually the last question I ask and it's interesting to see if it matches with how I rate them based on test questions.



Kyle Peatt

OCTOBER 14, 2013

This question is good if you follow up with "What made you choose that number?" This gives you a great chance to see exactly how familiar they are with a language because they'll be able to talk about the things they do know and what they feel like they don't know yet. Most good candidates will put themselves at a 7 or 8.



Senff

OCTOBER 14, 2013

Yup, I always do. It's not really a question you can answer correctly (most people give themselves 7 or 8 indeed) — but it IS one you can answer wrong, or at least make me wonder.

If someone says he's a 10, you know something's wrong. And yes, I've had people say they were. ;)



Mason

OCTOBER 14, 2013

As somewhat of a CSS beginner, I felt pretty good about knowing some of the answers to these questions! This also gave me some good learning topics to look into. Thanks for this Chris.



Michael Calkins

OCTOBER 14, 2013

I consider myself VERY educated with CSS/JS/PHP (AngularJS, Laravel) and I think these are excellent questions when looking for someone who is proficient in CSS.

The only thing I'd probably add is some sort of design question to try to weed out people who design like it's the 90's.

In your opinion what are some examples of good, usable, and engaging design?



Ferdy

OCTOBER 14, 2013

I agree with @Scott that whiteboarding really is a strange way to test somebody's skill, since it is so far from how you would typically work. I for one, am fairly experienced in many aspects of web design, yet I still have to constantly look up things on the web. In real life, that doesn't matter, as long as I get the job done, and I do. In a whiteboard situation, I would likely fail a lot. I am not against on-the-spot skill testing though, but let it be in a real-world scenario.



Ricardo Zea

OCTOBER 14, 2013

Exactly the same here.

Like you I'm fairly experienced but I find myself referring to my questions/answers in Stackoverflow,

my code snippets in Gist, and now I'm referring to my own demos in CodePen more and more.

If during the whiteboarding I am “allowed” to refer to my resources above, I wouldn’t have a problem with it.



alicetheblue

OCTOBER 14, 2013

gulp*

I am going for interviews and hope they don't ask me to do this.

But, wait. I will do this now. And then look uber smart. ;)

Thank-you Chris Coyier.



Catalin Rosu

OCTOBER 14, 2013

If I'd be in the position to interview someone, I would also add the following lines. It would be just fun to chat about :)

```
p[style="text-align: center"] {
    text-align: left !important
}
```



Billy

OCTOBER 14, 2013

I'm going to say that since you are referencing a paragraph which seems to have inline styling and knowing that inline styles are suppose to override everything, that this will still keep the text aligned center.

I guess the question is, does important override inline styles? or is inline styles the real deal lol



John

OCTOBER 15, 2013

No Billy, the text here would be aligned left. !important makes it precede the text-align center inline style, and because the style property is not interpreted as css, it wouldn't invalidate the selector.

**Billy**

OCTOBER 15, 2013

So important breaks all the css rules? meaning it can override anything?

I guess this is why its not good practice.

**Cesar Lee**

OCTOBER 14, 2013

The premise here is that "...you're in the position of needing to interview someone about their skill and knowledge about CSS...", so hell yes, I agree that these questions are adequate and well thought. Cheers!

**Billy**

OCTOBER 14, 2013

Hey Chris,

Thanks for this. You started out a little brutal lol the create the button question is not bad if you are allowed to sort of peak at things. For example I know exactly how to recreate that button carefully and which css properties to use, however remembering the syntax for things I don't use daily like box shadow or text shadow (although its in the form of 'property: x y blur'). Would be no problem with say a code editor like sublime to help with hints.

Sadly depending on text editors hurts you in the long run imo but with the push for "speedy development" we become reliant on tools to do things for us. Its a trade off, you don't focus on little details like this knowing that you have tools to help, in order to speed up your workflow.

Anyhow most of these questions are good questions. Regarding the button question, a good compromise would be to have a person DESCRIBE how they would make that button. I really liked the question about the selectors having to describe something like [role='xyz'] > a:not etc

**Kyle Peatt**

OCTOBER 14, 2013

Hey Chris!

Mobify Design Team Hiring Person here — thanks for featuring our pen! We've got a few other private pens that we also use to suss out peoples comfort and abilities.

For the people commenting on the value of CSS tests, we like to use them for one very specific purpose — although we get some other stuff out of it as well.

We administer them during interviews and like to sit in the room with the person as they try to figure them out. The main thing we're looking for is the ability to problem solve so we'll usually ask questions as they go about what they're trying and why they think it will work. This button one is awesome for both testing comfort level with some more advanced CSS3 stuff along with the ability to closely match a given design. We totally let people Google or ask us about properties because that's how they'd work anyways.

And if people think they'd nail this test or anything else we'd throw at them, we're constantly hiring developers and designers at <http://www.mobify.com/jobs>.



Kyle Peatt

OCTOBER 14, 2013

Also if you apply, I probably wouldn't expect to get this exact test anymore... Just sayin'



Kyle Peatt

OCTOBER 14, 2013

And my link didn't work properly.

<http://www.mobify.com/jobs>



Billy

OCTOBER 14, 2013

Sounds good to me. A simple cheat sheet of css3 properties would suffice and not as if its needed the entire time, but just to have it would be cool. The interaction is definitely important and also being able to expand on your thought process as to the why's etc



J.P. Berry

OCTOBER 14, 2013

I like the clearing floats question. I typically draw out two 50% width floated boxes, with a wrapper around them that has no height. I ask what would happen if you added a border to the wrapper, and why. So instead of asking "how do you clear a float?", I'm looking to see if the developer even knows when he/she needs to contain the floats in the first place.

I also like to ask the difference between using translate vs. position:absolute. I don't necessarily mention

animation in the question if the developer is interviewing for an intermediate / advanced position. I've seen some interesting answers there.

Lastly, I like to ask slightly more abstract questions that don't have a right or wrong answer to see a candidate's thought process. Something like, "Why are manholes round?". The answer itself doesn't matter, as the candidate (probably) won't be installing manholes, but it often yields surprising results that show both thought process and the willingness to think through a problem. It's also kind of a fun diversion from the often anxiety-producing technical questions.

**Billy**

OCTOBER 14, 2013

Kyle Peatt do you guys have remote positions?

**Kyle**

OCTOBER 14, 2013

Hey Billy,

We don't currently have any remote positions; however, we will explore relocation! Vancouver is an amazing place to live and work.

Kyle

**Billy**

OCTOBER 14, 2013

Ha, Thanks Kyle but I have already relocated a few times. I'm back home in NY or at least the northeast for now.

I'm engaged to be married, can't just up and move no more

**Matt Schiffman**

OCTOBER 14, 2013

Thank you Chris, for keeping me humble.



Ricardo Zea

OCTOBER 14, 2013

With all due respect, the first thing I thought I would say in an interview when I saw the example of:

Explain to me what's going on in this CSS selector:`[role=navigation] > ul a:not([href^=mailto])`

... was: "Dude! Wow, you're in trouble there bud."

Whenever you get the point of writing a monster selector like that, is because you're in DEEP sh#@!+. >_<

Explaining selectors is technical, anyone can study and memorize certain structures for the interview.

Is the analytical process and understanding the type of message that bad selector is sending out that I think is most valuable. At least, I think, it helps generate an opinion that could give the interviewer an insight of the candidate's knowledge and most importantly, experience.



Joey Iny

OCTOBER 14, 2013

Would CSS-Tricks itself be considered a valuable resource in these types of interviews?



Sebastian

OCTOBER 15, 2013

If you're smart, you won't play school exam during an interview. It's hard enough to find talented people and you don't want to lose them. And as mentioned before – nobody is looking for a textbook on CSS to hire, but on someone with a strong analytics skill and creativity to solve problems in an efficient way.

On a side note: The mention "if they know it, they can talk about it" is wrong. It's as wrong as it gets. There are tremendously gifted people who do extraordinarily great work – but when asked to explain, they terribly fail. We all know situations in life when we get asked to explain something but can't: "Well, er, it's hard to explain."



Jose Vega

OCTOBER 17, 2013

Sebastian, I totally agree with you. Some people are not good talking or explaining their ideas, but that doesn't mean they're not talented.

I often find myself in that position. Ideas and problem solutions flow in my head freely, but I don't find the right words to explain them to my co-workers.



Vijay Shinde

NOVEMBER 27, 2013

Sebastian, I totally agree with you. Some people are not good talking or explaining their ideas, but that doesn't mean they're not talented.

I often find myself in that position. Ideas and problem solutions flow in my head freely, but I don't find the right words to explain them to my co-workers.



Lamin Sanneh

OCTOBER 15, 2013

I think there's nothing wrong with this format of questions as long as you have a backup question in case they don't know how to tackle this exact problem. I mean come on, you might miss an extremely talented frontend developer like myself (I'm just kidding, just a cocky joke) just because of a simple edge case they couldn't solve on the spot. I am sure most of us have that one problem which is very basic, one that we know we should be able to solve but we often need some googling to get it done. What if it happens to be one such question for a developer.

It makes sense to ask research type questions. Ask them about their workflow, how they would solve a particular problem (one which they chose and thought was challenging). You could even ask them what they think was challenging about it. But don't take it word for word, we as front-end developers are problem solvers (whether it be googling, referencing a book, or plain old fashion remembering/memorizing css syntaxes).

In short, there is no hard and fast rule. I'd hire a better researcher than someone who is very good at what they do but can't get out of their comfort zone to do newer stuff as it's the nature of our field (things change quickly and constantly) and I like it that way.

Thanks for reading (Yes I know it's a bit long but I thought I had to make my feelings known, :-)).



Vladan

OCTOBER 15, 2013

You can show them a website (some of your finished project), and ask them how some element can be coded and styled. You can start with some simple things, and raise difficulty levels. In that way you can tell how far every of them can go, and what are their limits.

BTW, excellent post. I had some of those questions on my job interview.



Hardik

OCTOBER 15, 2013

you can find more programming tutorial here

[How to Read Image Text Using PHP](#)



fbalboa

OCTOBER 15, 2013

Chris rulz !!!!

Thanks, this is excellent for juniors front end devs



Niels Matthijs

OCTOBER 15, 2013

I've stopped asking technical questions about front-end (css) when doing job interviews. I'm more interested in how they pick up new techniques and how they look at the job (ie, will they last for a couple of years and will they continue to grow). Technical knowledge is something the right people pick up real fast when guided well, maintaining that knowledge and challenging themselves constantly is what we are looking for in good front-enders.



Lamin Sanneh

OCTOBER 15, 2013

Hi Niels,

Couldn't have worded it better. Staying current and being able to keep up with the industry (when necessary) is more valuable than stagnant skills.



Hugh Grigg

OCTOBER 15, 2013

Man, I used to think I was pretty crap at CSS, but having read through this I feel like I could have at least made a good effort on each of the questions / tasks. Before this I didn't think I'd be even slightly near the level required to use CSS professionally.

**Chris**

OCTOBER 15, 2013

I would definitely include a question about positioning and the position property. Something like “How do you center an absolutely positioned element?” would be very telling.

**Mark**

OCTOBER 15, 2013

I Googled that, problem solved in about 30 seconds, onto the next one :)
<http://www.sitepoint.com/css-center-position-absolute-div/>

**Rick**

OCTOBER 15, 2013

Excellent article. I appreciate the interview questions.

**dazzle89**

OCTOBER 15, 2013

Hello guys,

I tried to recreate the button, using an a-element only, without spans or anything else. Here is my solution:

<http://jsfiddle.net/UPJfd/>

But I have no idea how to insert the stars. Does anyone have an idea?

**Kevin**

OCTOBER 15, 2013

Hey that's a good duplication! The stars are the easiest part. Just use pseudo-elements like this:

```
a:before {  
    content: '★';  
    color: #999;  
    font-size: 1.1em;  
}  
  
a:after {
```

```
content: '★';
color: #999;
font-size: 1.1em;
}
```



Alex Wright

OCTOBER 16, 2013

@Kevin

If you're going to have two of the same rules like that, I would combine them with the same selector

```
a:before,
a:after {
    content: '★';
    color: #999;
    font-size: 1.1em;
}
```

@dazzle89 Another solution for the stars could be adding the actual html special character in the button. Yes, it's adding some more markup, but it will have great support since you're not using :before / :after.

[your fiddle, extended](#)



Kevin

OCTOBER 16, 2013

@Alex thanks for the pointer, definitely slipped my mind.



S

OCTOBER 15, 2013

Love it. Hot drama about a button, no-one see's the bigger picture.

— S

**Ricardo Zea**

OCTOBER 15, 2013

Whiteboarding? Sure, bring it... let's dance.

Wanna talk instead? No problem, I'm bilingual, let's both learn then.

Let's man-up already.

**christos**

OCTOBER 19, 2013

+1 :D

**Kevin**

OCTOBER 15, 2013

I did the checkout button in 30 minutes... am I still hired? <http://codepen.io/gillytech/pen/duHFG>

**Kyle Shevlin**

OCTOBER 15, 2013

Kevin, great pen. I've been looking at it because I knew there would be parts I would have difficulty with on the button. I think there's just one thing that would make it even closer. I find that `a {letter-spacing: -1px;}` seems to set the kerning correctly on the letters. They are a bit closer together in the example than in your pen. What do you think?

**Kyle Peatt**

OCTOBER 15, 2013

Hey Kevin,

Nice work on the button! It's pretty close although I see a couple of minor differences you could touch up. Awesome work though.

Can't hire you on the spot but feel free to apply on our jobs page.



Kevin

OCTOBER 16, 2013

Thanks guys. I made a few edits and I'm pretty sure this is a spitting image.

I shortened the letter spacing as Kyle S. suggested and added a shadow around the button that I missed. Also changed the main gradient to more closely match the original

Thanks for the offer Kyle P. but I'm just honing my skills for my current job :)



Kyle Peatt

OCTOBER 16, 2013

Hey Kevin!

Looking good. Missing one thing, I think. Check out the shine on the button a little bit more closely near the left and right edges.

Thanks for playing with our button. Might be fun to open up a few of our other tests.



Kyle Shevlin

OCTOBER 15, 2013

Thank you for this post. Teaching myself web design/development for a year now and am trying to get my foot in the door in this career field. I have a feeling that practicing some of these things might finally get me hired.

hire.kyleshevlin.com



Joseph Levin

OCTOBER 15, 2013

Loved the article. Pure gold as always. Thanks for posting it!

I am curious about the CSS button task:

Is the image shown generated from an image editor (like Photoshop), or was it already generated by CSS? (sneaky, sneaky!) If it was already generated ONLY from CSS, how were the small curves created that occur along the 50% point of the linear gradient to the right and left sides of the button?

Also, is there any way to get the outer box-shadows to respond to hover as the main, central portion of the button does normally (without resorting to extra markup or using JS)?

In my pen, what I'm calling “The Interview’n Button”,

I would also love to know, should a small nudging margin be placed on the a:hover, why the star glyphs get slightly larger.

Just wonderin'.



Kyle Peatt

OCTOBER 15, 2013

Hey Joseph,

What you see was generated in browser! Yes, including those small curves you mentioned. Not going to give too many hints but you can do it with one of the properties you've used already.

As for the having box shadows react to hover, you might be able to use the same technique required above to get something like that working.

Thanks for playing with our button!



Joseph Levin

OCTOBER 15, 2013

Kyle,

Yeah. I just didn't think deeply enough about it at that moment. Those curves to each side are actually simple. Plus, you can then remove the linear gradients for a slim-down, which is an extra nicety. Here's the "The Lighter, Tighter Interview'n Button"

I might have worked-in an extreme-sized radial gradient to cause the entire line to have a slight curvature ('cos I think that looks better), but the goal was to emulate the example.....

It might be nice (hint, hint) if Chris or others could post more things like that for us to emulate and then show some interesting responses.

I mean, I've often seen an interesting nav or interface and then done my best to emulate it, as an intellectual exercise. Sometimes that can be a challenge, but almost always fun to do.

Thanks for your kind response, BTW.



Matt Bain

OCTOBER 15, 2013

Tests aside, my go-to CSS questions are:

can you explain CSS specificity?

please explain how CSS Selectors are matched by browser engines?

**Jelmer**

OCTOBER 16, 2013

Also:

Name some of the news websites/blogs/resources that you visit on a regular basis to stay up to date with web technologies.

Or

Name some of the people you follow online (Twitter/blog posts/RSS/...) that are important for continual improvement of your development skills and/or the web.

Bonus question:

What are your favorite GitHub repo's?

:)

**Ricardo Zea**

OCTOBER 16, 2013

I totally disagree with this question:

What are your favorite GitHub repo's?

Don't get me wrong, I'm a big fan of Github (who isn't really) but honestly I don't see the relevance of how many "cool" repos a person knows about or follows or whatnot. Hell, all someone needs to do is say: "Yeah, jQuery, HTML5 Boilerplate and Emmet". So what, is the person now all of a sudden fit for the job?

If the person uses Github or at least they've tried to use it, now that's a route with more value.

**John**

OCTOBER 17, 2013

Jelmer – Aren't those questions straight from Darcy Clarkes front end interview techniques?

I work with some great developers who don't use Twitter or Github and it doesn't stop them from being terrific at their job.

It's more important to try to find out if the person is a good fit for your team/company rather than who they follow on Github or Twitter.

**jimpogo**

OCTOBER 17, 2013

This article made me realize some gaps in my own knowledge! Yikes. Thanks for the summary.

My company's user base tends to use older browsers, so we'd have to add more on fall backs.

**Richard**

OCTOBER 18, 2013

Great article (and some interesting comments) Chris. It's a simple fact that many of my students are required to undertake a test as part of their interview process when applying for front-end roles. One company in Manchester was asking applicants to 'draw a pair of pants in CSS'. Tests seem common practice. I'd say the article provides a lot of food for thought on both sides of the table. Thanks.

**Gilberto Crespo**

OCTOBER 18, 2013

Hey Chris, I'm your big fan and I really liked this article. The content seen a little bit extensive to apply in an interview, BUT opened my mind about the importance to update the content.

In addition, I would tell you it may be applied for current workers who **should know** these techniques and it isn't what always happen, and making easy the tracking for further training.
Thanks.

**christos**

OCTOBER 18, 2013

disclaimer: i haven't read every comment.

all i can say about the ones i have read though is, what's the big deal?

i'll do a test on a white board, piece of paper or computer, why? because i know what the fudge i'm doing. it's not insulting at all, probably no more insulting than bringing an attitude over testing methods to an interview where you want these people to pay you money to do a job for them.

if you want to complain about a test you're given, first blitz the thing and then tell the people how it could be done better, constructively.

also, when i have to interview someone and they don't come with a showcase, either on their laptop or a url to their site showing previous work, then there's 99.999% chance they're not getting the job, it's not that hard.



Ricardo Zea

OCTOBER 18, 2013

You should read my comment here :)

Spoiler:

110% agree with you.



Billy

OCTOBER 18, 2013

I would prefer someone looks at my previous work but I'd be ready to interview to answer questions or code up something.

One thing I don't like though is when companies ask you to do things you probably won't do at the job. For example, if they have no buttons that look like that on any of their websites/ work then why? lol Test me based on what you need me to do for you. Another example is asking me about say animations when you don't use them



sheriffderek

OCTOBER 18, 2013

My coworker didn't read the whole article and thought it was a contest. I got pretty close I think. [CODEPEN](#). I think this is a great way to vet people and see how they think. I just ran some potential employees through some tests and I found it to be extremely revealing. I mean, if you can't copy a button, then maybe you aren't that great at writing CSS. Maybe you would be better with content strategy or other aspects of the job. You have to really love CSS if you are going to write it all day. I would think that people who love CSS would be happy for the challenge.



Geoffrey

OCTOBER 19, 2013

[role=navigation] > ul a:not([href^=mailto])

While good interview material and test of css syntax... a good follow up question would be if they would ever try and use something like that.

i.e. <http://css-tricks.com/efficiently-rendering-css/>

**Billy**

OCTOBER 19, 2013

It depends. Yes using long css like that is probably not efficient but its probably better for maintenance.

You can put a class on every single link for mail to which might not be alot in this case, but what if you want to have certain links be a different color than all the normal colors? for example external links vs internal links?

You can add a class to every single link but that adds to markup. I feel like they both have their pros and cons as with most things. Another example is, every time you add a new link that fits that scenario you have to remember to add the class vs adding it inside the ul it will automatically be picked up.

So why it may be no good for a certain business who say cares more about performance than another, everything should be situational.

**Samiullah Khan**

OCTOBER 20, 2013

As far as those goes, **chris** you didn't asked much about the advanced or how much we've to prepared for **CSS** animations and what about *CSS Transform*. And more Importantly the new layout system like **Flexbox** should we worry about those things as an interviewee?

This comment thread is closed. If you have important information to share, you can always contact me.

What now? I have some ideas for you.

We can do the real footer now.