Testing HTML/CSS/Javascript skills when hiring [closed]

Asked 16 years, 2 months ago Modified 10 years, 1 month ago

Viewed 20k times

27

Closed. This question is opinion-based. It is not currently accepting answers.

Want to improve this question? Update the question so it can be answered with facts and citations by editing this post.

Closed 11 years ago.

Improve this question

When hiring a front-end developer, what specific skills and practices should you test for? What is a good metric for evaluating their skill in HTML, CSS and Javascript?

Obviously, table-less semantic HTML and pure CSS layout are probably the key skills. But what about specific techniques? Should he/she be able to effortlessly mock up a multi-column layout? CSS sprites? Equal height (or faux) columns? Does HTML tag choice matter (ie, relying too heavily on `<div>`)? Should they be able to explain (in words) how floats work?

And what about javascript skills? How important is framework experience (jQuery, Prototype, etc). today?

Obviously, the details of the position and the sites they'll be working on are the best indication of what skills are needed. But I'm wondering what specific skills people might consider to be deal-breakers (or makers) when creating tests for candidates.

javascript html css frontend

Share

Improve this question

Follow

edited Oct 29, 2014 at 6:23

John Saunders
162k ● 26 ● 249 ● 402

asked Oct 13, 2008 at 17:26

Bryan M.
17.3k ● 8 ● 48 ● 60

13 Answers

Sorted by: Highest score (default)

▲

19

▼

When I interview people for a position of Client-Side developer I try to figure out:

```
1) Understanding DOM (what is that, how is it
related to HTML etc)
2) Understanding XML/namespaces
3) Understanding JavaScript (object-oriented? what
otherwise)
4) Knowing approaches to componentization (XBL,
HTC) - plus
```

```
5) Understanding OO principles
6) JavaScript closures
7) Memory leaks in browsers
```

The only simple test case I give right away:

```
<script type="text/javascript">
var a = 1;
</script>
```

I suggest interviewee to explain in technical terms that line.

And I also check on overall awareness of the current state of the Web technologies, among other questions I suggest designing a web-browser where interviewee is suggested to pick technologies he would put into his brand-new creature, suggest ones that might be missing from the current client-side platform.

Share Improve this answer Follow

edited Oct 13, 2008 at 19:43

answered Oct 13, 2008 at 17:33

Sergey Ilinsky
31.5k ● 9 ● 56 ● 56

1 typo, line 3: "Undesrtanding" (wish I had edit rep ...)
– Bobby Jack Oct 13, 2008 at 18:37

2 And what answer would you expect from the simple test? a is a variable declared with global scope with its value set to the numeric value 1? – Jacques Nov 6, 2013 at 5:11

7

I can suggest you online test, that includes HTML, CSS and JavaScript together.

http://tests4geeks.com/test/html-css-javascript

It has 60 questions (20 for each subject). And you will recieve the report via email, when the candidate finishes the test.

Share Improve this answer Follow

edited Apr 10, 2013 at 7:29

answered Feb 14, 2013 at 12:05

Vadim
131 ● 2 ● 2

5

Sergey and swilliams both gave great answers, in particular, swilliams mention of asking for a portfolio is key. With a portfolio you can also test for items like,

- does the html and css validate?
- does the presentation render consistent across browsers?

- does the candidate have JavaScript errors? if they do, does the person let them bubble up to the presentation layer or do they at least catch them with a try/catch block?
- in terms of JS, how advanced is the person? Can they do form validation? Can they do regex? Are they relying on MM_Preloader? (Yuck!)

A portfolio can also give a sense of how passionate someone is about web development. Moreover, if they've done a site for someone else, that alone presents an opportunity to talk about a number of things with a candidate,

- how did they go about developing the UI?
- what kind of planning went into the site?
- how were user expectations uncovered/met?
- what kind of challenges during construction came into play?

Beyond these items, one other approach you might want to consider is a developer test that you could send a prospective hire. Nothing too hard that would take more than a day, but enough of a brain teaser to see if they can work through a CSS or JS problem.

Share Improve this answer Follow

answered Oct 13, 2008 at 18:12

Anjisan

1,799 ● 3 ● 15 ● 26

4

Ask for a portfolio, and then review it with your team. That takes care of posers and people who "don't handle interviews well."

Other than that, I'd present them with something relatively simple to mock up and a laptop and say 'have at it.'

Maybe ask them what they like most about web design today, and what they hate the most. Ask them about their opinions about what is on the horizon (HTML 5, IE 8, Chrome, etc) to see if they keep abreast of what's coming out.

Ask them if they have a favorite JavaScript framework and why. Maybe have them code something in JS a la the [in]famous fizz buzz problem.

Share Improve this answer Follow

answered Oct 13, 2008 at 17:30

swilliams
48.8k ● 27 ● 102 ● 130

Asking for a simple mock up won't really get at the most important CSS skills, which are handling complex cases and multiple environments. Also, some people genuinely don't interview well (I am one of them, but my work makes up for it). – eyelidlessness Oct 13, 2008 at 17:40

That's what the portfolio is for... – swilliams Oct 13, 2008 at 18:40

▲

4

▼

At my company we don't work with tests and the portfolio is more important, especially because we tend to look at the candidate's personal motivation and passion for doing front-end development.

But if I should give the candidate a test before hiring, I would go about it this way:

Hand over a print-out of a photoshopped web page representing a clear semantic component tree underneath. Ask the person how he or she would come to a result in html. Just ask him or her to think out loud. What goes through one's head when seeing a page, knowing it needs to be developed.

Then it comes down to the approach the candidate takes.

Choosing the best available markup for each specific html section is one (important) aspect, but can be mastered with experienced people around to guide a new employee. Being able to properly break down a design into its semantic components, identifying sections and separating primary and secondary content from navigation is not math or science and therefore hard to test. But a conversation about the approach of breaking down a page might separate experienced people from beginners.

But as I said in my first line, we usually ask for what kind of web-related work a person has done in his or her free time, such as a blog, game or demo. If done anything, the person usually was really good at front-end development or was eager to learn and adapt.

Share Improve this answer Follow

answered Oct 14, 2008 at 6:10

Martin Kool

4,247 ● 4 ● 27 ● 36

Could you elaborate on what a "semantic component tree" is? Or point to any resource explaining it? Thanks :) – o01 Apr 27, 2011 at 7:33

2

We used to either set up a small brief for them to work within a certain timeframe, or in some cases contract a job to them and see how they went.

I was never that worried about making someone sit down at a laptop in an interview room and bang out a solution, because that kind of environment is not (you'd hope) very much like the normal working conditions.

The exact nature of the brief will depend a lot on the skillset you are looking for. In some shops, the front end developers will have to take on a certain amount of filling the gaps in design, and/or performing "brand police" duties as they implement the design "vibe".
In those cases, leaving some holes in the brief regarding some of the finer points of the typography and other small details can give you some hints as to their abilities in those respects.
Let them choose the javascript framework, did they choose the one best for the job, or the one that they

knew. (subjective question, yes. But pulling in dojo for the sake of a datepicker and some menu animations might be overkill)

I would look for someone who can implement css based layouts, but can in fact work on table based layouts if needed when dealing with legacy projects. Maligned as they are, some of the finer details of hacky table layouts weren't always easy.

The main thing in this sort of task is the attention to detail, have they added a set of style rules for printing, used appropriate image sizing and formats, produced clean and valid code, have they gone for gold because they really want the job, and are prepared to stretch themselves a little to get it.

Because you give them some extra time, it is fair to expect that they try to impress, rather than the stress laden scenario of putting them at a strange desk and telling them to go for it. So, whilst being realistic and not expecting obsessively perfect work every day on every task, in this situation I'm looking for gold, or at least evidence that gold was the target.

Throw in a curve ball of something they haven't done before... see how well they can pick it up in a hurry. Experience is good, but the ability to learn fast is probably more important in an area that changes so fast.

answered Oct 14, 2008 at 2:55

Share Improve this answer

Follow

seanb
6,954 ● 2 ● 34 ● 34

1

An understanding of how browsers are different is also key. Especially IE. If they have only ever coded for IE beware! Vica Versa too, if they've never tested their stuff in IE6/7 they have no idea how horribly it is failing.

If they can name 3 things that fail, or need workarounds for in IE, then they're likely on the ball. If they can't, they haven't had enough cross browser experience yet.

Share Improve this answer

Follow

answered Oct 13, 2008 at 22:03

scunliffe
63.5k ● 26 ● 131 ● 165

1

I ask people what tools they use, how they code, i.e. do they use DreamWeaver, BBEdit, emacs or whatever. Assuming they don't just give a one-word answer, you normally get some kind of idea what their priorities are, how they code, etc.

Then I ask how they validate their code, which is always interesting.

Aside from actually testing them by having them sit down and hammer out a page, I would ask them for specific examples of work they've done, how they've resolved an issue.

For instance, you say *"tell us about a job where you were on a strict deadline"* or you say *"tell us about a really frustrating problem and how you overcame it"* or *"in the work you've done recently, what's the thing you're most proud of?"*

That way you get a lot of insight into the kind of thing they've done, their problem-solving skills and experience, the way they handle stress and frustration, and of course, whether their workarounds/corner-cutting measures are smart or dumb.

Share Improve this answer Follow

answered Oct 14, 2008 at 2:18

AmbroseChapel
12.1k ● 7 ● 49 ● 69

▲

0

▼

Thanks for all you're answers so far. They're definitely good advice. I'm not ready to mark one as accepted though. What I was really looking for were concrete tasks that a front-end developers need to be able to produce in a test environment. In other words, what is FizzBuzz for CSS?

I do agree that a solid portfolio and the ability to speak intelligently about the subject will probably minimize the importance of a test. On the other hand, I think we all know candidates who are very good at talking big, but when it comes down to actually demonstrating practical knowledge, they fall short.

@JonathanHolland - I tend to disagree. HTML/CSS is certainly easily learned, but the knowledge gained from a few quick tutorials is nothing compared to someone who has spent years doing layout. There are lots of techniques and practices that a good CSS dev can bring to the table.

@roenving - Perhaps I was redundant. By "pure-css", I meant table-less. I also meant table-less in the sense of not using tables as a layout methodology. However, this is not the question for a div-vs-table layout discussion. Try here. Although I believe that most employers these days are going to be asking if a dev can do layout without the use of tables (as a methodology).

Share Improve this answer

Follow

edited May 23, 2017 at 10:30

Community Bot

1 ● 1

answered Oct 13, 2008 at 22:04

Bryan M.

17.3k ● 8 ● 48 ● 60

Ah I guessed so, but the reality of my answer is that these words have been said and printed so often that a lot of new developers think that tables and css are two things that can't be combined ... And that is a problem (just think, there are css-properties that only works for tables !-) – roenving Oct 13, 2008 at 22:24

0

I don't have any experience with hiring, but I've attended a number of Web Developer interviews and can relate my experiences.

Out of all my interviews one method really stuck out as the best way of discovering true talent. I'll admit that I'm no expert, and this is probably why I didn't get the job, but this was a fantastic way of weeding out those that were capable and those that were great.

We were asked a couple of questions about DOM, Object-Orientation, Closures, XML Namespaces and general Web Design, then we were taken into a quiet office with a nice view across the nearby river and asked to write a few examples to show our ability. We were left alone, but were told that the source code and our browser history would be viewed after, if we needed to use a browser. We were asked to make a company intranet using the following:

- A simple three column layout
- A good-looking table using CSS
- A usable but good-looking Navigation Bar.
- Dynamic loading using XML and JavaScript

Given a small amount of time to do this, you get to see people that can really do this stuff. Whilst perhaps their example could always be better or more suited to different applications it just goes to show that the best way to show the skill of a developer is to **get them to**

make something! If you know your stuff and they know their stuff their code will be good.

Share Improve this answer

Follow

answered Nov 20, 2008 at 15:38

Mike B

12.8k ● 20 ● 84 ● 109

-1

"

Obviously, table-less semantic HTML and pure CSS layout are probably the key skills.

"

I don't understand that sentence ...

Do you really mean that it is better to do what is a simple task using tables in a complicated manner just to avoid using tables ?-)

Table-o-phobia is just as hard a disease as making large-scale websites without serverside assistance ...

Of course the table-hells of the last decade isn't interesting, but a lot of tasks is really stupid to complete without using tables ...

Use the html-element which easyist completes the task, no matter which tagnames it uses !-)

-- and I don't understand what a 'pure css-layout' is; I never figured out how to create an html-page without html-elements to host the layout in the stylesheet ...

Share Improve this answer Follow

answered Oct 13, 2008 at 21:55

roenving
2,626 ● 15 ● 14

Using a table for layout is semantically wrong because the tag conveys the notion of tabular data. The proper solution is to use divs, which convey no semantics, and to set their display properties to mimic tables. Unfortunately IE doesn't support display: table-cell.... so sometimes you're stuck. – Mr. Shiny and New 安宇 Oct 28, 2008 at 13:10

1 What's the idea in avoiding tables and then force other elementssto act like table-cells ?-) That really is semantically wrong ... As said above: Use the html-element that fulfills your task, and saying that tables only should be used for tabular data is not knowwng the history of the web !-) – roenving Oct 28, 2008 at 16:07

1 I'm not sure you'd actually land a job with this opinion. Just sayin'. – miketaylr Dec 4, 2009 at 14:49

-2

I know this isn't a great answer, but one of the first things I do is expose the formatting of their Word resume. If they've made use of the indents and styles that Word offers and it's not tab, tab, tab, space, space, space, space then they go top of the pile.

Share Improve this answer Follow

answered Oct 13, 2008 at 19:43

Steve Perks
5,540 ● 4 ● 30 ● 41

1 I like formatting stuff with the use of css, but seriously,word? that has absolutely nothing to do with front-end skills. – gion_13 Jun 3, 2011 at 13:46

1 I actually like this answer. It has to do with an attitude. it's like using `!important` instead of figuring out why something doesn't do what it's supposed to do. It works, and it's okay during development. But you should clean up, and present something that's build along some standards so it's easy to update/change. – Pjetr Jul 4, 2013 at 15:05

-9

I wouldn't put too much weight on it, as proper HTML/CSS is so simple that anyone can learn it in a week.

That said, you might want to review their portfolio to help you make a decision on their current skill level.

Share Improve this answer Follow

answered Oct 13, 2008 at 17:30

FlySwat
175k ● 75 ● 248 ● 314

2 HTML/CSS is simple if you don't care about the quality of the markup or the ability to make complex designs work on a broad range of platforms. I don't want to hire someone like that. – eyelidlessness Oct 13, 2008 at 17:38

It IS simple, but there are so many devs out there that still suck at it. – swilliams Oct 13, 2008 at 17:39

If you can't learn "GOOD" HTML/CSS design practices in a week, I wouldn't hire you either. – FlySwat Oct 13, 2008 at 17:49

3 The standards can be learnt in about a week, but it takes a lifetime to learn how vendors have misinterpreted those standards. – Bobby Jack Oct 13, 2008 at 18:39

1 It is simple, but it can just as easily be misapplied - no-one working with CSS and HTML after a week for the first time will be producing competent work if you want it to be valid, semantically correct, maintainable, working cross-browser, etc, etc. – FinnNk Nov 23, 2009 at 14:28