

REQUEST FOR COMMENTS: JAVASCRIPT SKILLS ASSESSMENT

CREATING AN OBJECTIVE JAVASCRIPT SKILLS ASSESSMENT TOOL

I would like to propose a model for a JavaScript skills assessment tool that will allow the individual to objectively gauge his/her skill level on a 1-to-5 scale. There are a number of reasons for building such an instrument beyond needing a good excuse to develop a new web app, including:

- It would provide novice developers with a map to help navigate the ordeal of learning professional JavaScript programming.
- It would provide feedback to guide workshop planning, thus helping create a more rewarding experience for all who attend.

To that end, I have worked out a conceptual model which I describe below for your consideration and comment.

SKILL-LEVEL ASSESSMENT BEGINS WITH IDENTIFYING THE CORE SKILLS, RIGHT?

It seems a no-brainer to say that you build a skills-assessment tool by identifying the core skills a developer needs for mastery, and then you create a test around them. That's how I started, and when I had listed north of 100 specific, discrete programming skills it became clear that it's not that simple. Because JavaScript development evolves so rapidly, getting a useful measurement is like hitting a moving target. Linking an assessment tool *directly* to specific skills -- say something like the ability to write code that operates on both IE 10 and IE 6 -- makes the tool obsolete as soon those skills become irrelevant.

What seems to be needed is a framework that represents the changing universe JavaScript skills as a relatively stable set of *abstract skill areas*. Each skill area could then represent a related but evolving set of specific skills. Over time, new skills could be added and others removed without rendering the framework itself obsolete.

Here then is a list of 10 skill areas which I think is fairly comprehensive and stable, offered to the group for comments and suggestions.

1. **JavaScript core language programming basics**
2. **Working with browser extensions, HTML, CSS, and HTML API's**

3. JavaScript Object Oriented Programming (OOP)
4. Using JavaScript development tools and related technologies
5. Employing JavaScript libraries and frameworks
6. Developing with JavaScript on the server and in enabled devices
7. Working with professional development standards and unit testing
8. Implementation of advanced JavaScript design patterns and web services
9. Managing platform compatibility, performance measurement, and optimization
10. Professional JavaScript application development and security

AN EXAMPLE OF A SKILLS AREA WITH SOME SPECIFIC ASSOCIATED SKILLS

As a way of illustrating this concept, the JavaScript Core Language Programming Basics skill area could be comprised of these specific skills:

- Create and include scripts in a web page
- Use variables, statements, and loops
- Create and use array data
- Create and use functions
- Have a basic understanding of Scope
- Understand and work with Objects and object Methods
- Access, validate and submit form data
- Access and manipulate the DOM
- etc..

MAKING THE FRAMEWORK USEFUL

Assuming that this framework concept is sound, the next question seems to be, *how do you implement it in a useful way?* If you look at the skill areas that I've listed, you may notice that they're in something of a sequential order -- fundamental skills first, leading to progressively more advanced topics.

As many know, however, learning a programming language is not a linear process. Students begin with different experience levels. They tend to skip from topic to topic driven by their interests or the need to solve specific problems, and most spend their careers filling the gaps in their understanding of subjects they've "mastered" long ago.

Consequently, layering a 1-to-5 scale on top of a skills-area framework requires a little finesse. A useful assessment tool needs to measure more than just breadth of understanding (spanning skill areas). It should also consider one's depth of understanding within any given skill area. I propose using a four-tier scale to assess depth of understanding: (1) Learning, (2) Familiarity, (3)

Competency, and (4) Mastery. (I know this is getting confusing, but stick with me it will all be clear in a minute.)

PUTTING THE PIECES TOGETHER

For the sake of illustration, let's say you're given a self-assessment form that asks you to rate your aptitude on a range of concepts drawn from across the various skill areas. Based on your answers, this tool would calculate your depth of understanding for each skill area using the scale like this:

- **>= 0%** Learning **< 50%**
- **>= 50%** Familiarity **< 75%**
- **>= 75%** Competency **< 90%**
- **>= 90%** Mastery **<= 100%**

These depth and breadth measures, in turn, would be used to objectively calculate your skill level using a 1-to-5 scale, where each skill level is defined by its minimum or baseline requirements. For example:

JavaScript Level 1, Baseline Requirements	
Skill Area	Requirement
Willingness to Learn	

JavaScript Level 2, Baseline Requirements	
Skill Area	Requirement
JavaScript core language programming basics	Competency
Working with browser extensions, HTML, CSS, and HTML API's	Competency
JavaScript Object Oriented Programming (OOP)	Competency
Using JavaScript development tools and related technologies	Familiarity

JavaScript Level 3, Baseline Requirements	
Skill Area	Requirement
JavaScript core language programming basics	Mastery
Working with browser extensions, HTML, CSS, and HTML API's	Mastery
JavaScript Object Oriented Programming (OOP)	Mastery
Using JavaScript development tools and related technologies	Competency
Employing JavaScript libraries and frameworks	Competency
Developing with JavaScript on the server and in enabled devices	Familiarity

JavaScript Level 4, Baseline Requirements	
Skill Area	Requirement
JavaScript core language programming basics	Mastery
Working with browser extensions, HTML, CSS, and HTML API's	Mastery
JavaScript Object Oriented Programming (OOP)	Mastery
Using JavaScript development tools and related technologies	Mastery
Employing JavaScript libraries and frameworks	Mastery
Developing with JavaScript on the server and in enabled devices	Competency
Working with professional development standards and unit testing	Competency
Implementation of advanced JavaScript design patterns and web services	Familiarity
Managing platform compatibility, performance measurement, and optimization	Familiarity
Professional JavaScript application development and security	Familiarity

JavaScript Level 5, Baseline Requirements	
Skill Area	Requirement
JavaScript core language programming basics	Mastery
Working with browser extensions, HTML, CSS, and HTML API's	Mastery
JavaScript Object Oriented Programming (OOP)	Mastery
Using JavaScript development tools and related technologies	Mastery
Employing JavaScript libraries and frameworks	Mastery
Developing with JavaScript on the server and in enabled devices	Mastery
Working with professional development standards and unit testing	Mastery
Implementation of advanced JavaScript design patterns and web services	Mastery
Managing platform compatibility, performance measurement, and optimization	Mastery
Professional JavaScript application development and security	Mastery

YOUR OPINIONS AND SUGGESTIONS WILL HELP

I would appreciate your comments and questions. Also, if you have a better idea, or if you see how I have gone completely off the beam, please speak up. Thanks in advance for you email responses to riverside.jsw@gmail.com.

Sean Olson
5/11/2015