

How mad are your skillz? Each category earns up to three points. You earn the number of points that represent your highest level in each category.

## Good person skillz

Up to...

- +10** if you are a good communicator
- +20** if you are a cool person
- +10** if you are a cooperative team member
- 20** if you are an @55h0I3

## HTML

- 1** Aware of markup language and how it's used; able to view source; can understand tableless layout
- 2** Write valid markup by hand, understand cross-browser testing and compatibility; use debugging tools
- 3** Strong web standard awareness; know specific rendering differences, familiar with emerging technologies (HTML5, for example)  
Portfolio: Show pages of clean markup that validate. Implement HTML5 features and demonstrate.

## CSS

- 1** Know how CSS affects style of page(s) across site; understand "cascading"
- 2** Write valid cross-browser-compatible .css files by hand; use debugging tools
- 3** Understand CSS3 and preprocessors; aware of emerging technologies; understand performance and caching

## Javascript

- 1** Know what it's purpose is and that it's not Java
- 2** Can traverse the DOM; use at least one library; can hook into events
- 3** Familiar with emerging technologies (Harmony, for example); active in the JS community; able to implement web sockets

## PHP

- 1** Understand that it runs on a server and is a programming language; can describe dynamic vs static web pages

## PHP (continued)

- 2** Understand basic programming concepts (variables, arrays, functions, control structures); can read code and edit it; can perform single-line database queries (select, insert, no joins or subqueries); understand OO enough to use classes and objects
- 3** Write OOP, use libraries, apply big O (attempts to write code that runs quickly), code is elegant

## Databases

- 1** Can create databases and users using GUI tools
- 2** Can execute queries; create tables; write semi-complex queries (one join)
- 3** Can tweak indexes; architect performance optimizations; write complex queries with multiple joins; implement transactions; can rewrite slow performing queries; experience beyond MySQL

## DevOps

- 1** Can use a version control system for basic tasks; separate dev environment from live
- 2** Use version control on a team project (branching), set up dev->stage->live environments; follow QA process
- 3** Use feature branching; automated deployment; automated testing

## Sysadmin

- 1** Understand major Linux package managers and can add/remove packages; configure repositories; perform common command line tasks; configure components of the LAMP stack
- 2** Know the purpose of each folder in a standard Posix system; can inspect and configure network devices and firewalls; create backup strategies; basic security; can setup a new server
- 3** Automatic configuration/deployment tools such as bcfg2; advanced security knowledge; can use penetration testing tools, set up intrusion detection tools.

## Performance and scalability

- 1 Understand how clusters of servers can be load balanced; understand the limitations of and can work with replicated databases;
- 2 Write code that employs "just enough" caching; profile a site to discover where performance bottlenecks are; know Varnish and memcache
- 3 Can use lower level OS and system profiling tools such as sar, oprofile; can design a scalable architecture and predict problems; knowledgeable about cutting edge technologies such as NoSQL databases (MongoDB, Cassandra)

## Drupal site building

- 1 Can create users, content types, fields; create roles and permissions; create blocks administratively
- 2 Can install Drupal, add modules and themes (in the right place), create views, and use panels
- 3 Use features, apply rules (actions and layouts), implemented complicated module integration or views

## Drupal development

- 1 Can apply a patch; understand "hooks"; can read module code
- 2 Can use the form API, hook menu, and database API
- 3 Can write custom modules using Drupal best practices; write custom drush commands or cron tasks; write performance-enhanced code

## Drupal theme development

- 1 Install a drupal theme, customize settings via UI, modify templates, make cosmetic changes to Drupal's default output
- 2 Use base theme / sub theme, create custom templates, apply advanced front-end concepts like accessibility, responsive layouts.
- 3 Create a theme from scratch, use template.php to the fullest by using or creating theme hooks to modify Drupal output, can add theme functions to modules

## Visual design

- 1 Design-software use knowledge (Photoshop, Illustrator etc); know visual hierarchy of elements in design; designed static websites

## Visual design

- 2 Designed logos, brand elements and style guides; design documents are organized and clearly labeled; designed for dynamic websites (CMS)
- 3 Have a library of design tools and styles that are used to rapidly create elements; can transform own artwork into markup with platform specific conventions; design states of inter-active elements

## Information Architecture

- 1 Understand the importance of organizing all pages/content at the onset of the project because IA influences navigation, interface, interaction, and visual design (foundation)
- 2 Use the personas and goals of the site to clearly illustrate the relationship of pages in order to provide workflow diagrams and logical site navigation (framework)
- 3 Understand how aspects of IA/navigation can be translated to a specific technology (Drupal), experience implementing modules that aid in this process (whole house)

## User Experience

- 1 Know the difference between a site map and wireframes; can build basic wireframes
- 2 Build user flow / process flow diagrams and detailed wireframes with illustrated interaction patterns; aware of recent trends and accessibility guidelines; consider major aspects of accessibility (visual/hearing impairment) when designing a site
- 3 Perform user testing of all levels, expert knowledge of accessibility regulations / guidelines, consider many levels of accessibility (color blindness, etc) when designing a site

## Open source involvement

- 1 Attend local meetups; use issue queue; edit doc pages
- 2 Volunteer at events; help on IRC; contribute patches, small modules or themes; propose sessions
- 3 Organize events; become a community leader; maintain major modules, forks, or distributions; contribute to core

## How'd you do?

-20 - 28

BEGINNER

29 - 57

INTERMEDIATE

58 - 85

ADVANCED