General Stuff

## What Happens After URL is Entered?

DNS lookup

* Try browser cache, then os cache, then router cache, ISP DNS cache
* Then if those fail there is a recursive search from the root name server

Initial GET request

* Keep TCP connection open
* Contains cookies in the request that have info for user

Site responds with permanent redirect ([www.asdf.com](http://www.asdf.com) instead of asdf.com)

Browser follows redirect with another GET request. Server handles the request and sends back a response with encoded HTML

HTTP/1.1 200 OK

Browser then renders HTML and sends more GET requests for any tags that link elsewhere

## What are the main components of an HTTP request?

1) Request method, URI, protocol version

2) HTTP request headers

3) Empty line, then HTTP request body

## Regular Expressions

Text-based engine, regex-based engines. Mostly regex-based. Leftmost matches, regex engine tries all permutations of leftmost characters until match.

**^ … $** start and end anchors for regex

**[^…]+, [.]\*** anything but character ^ and everything character \*

**[a-z0-9A-Z]** character sets

**\d** = digit, **\w** = word, **\s** = string

**\D** = [^\d], and so forth

**?** makes preceding character optional

**{2,}** matches 3 characters or more (to infinite)

**{2,4}** matches between 3 and 5 characters

## JSONP

JSON with “padding”, it provides a method to request data from a server in a different domain. Basically what this means is you can request JSON from a server and it comes wrapped in a function call so that a function **that’s** already defined in the JS environment can manipulate the JSON data.

Said differently, the typical use of JSONP provides cross-domain access to an existing JSON API, by wrapping a JSON payload in a function call.

## SQL Syntax

If there are a bunch of customer transactions and you want to get the sum of each individual customer, you would group by customer and select customer and SUM(OrderPrice). If you wanted to specify that further, you would use Having. For example, if you wanted transactions less than a certain amount you would do

SELECT Customer, Sum(OrderPrice)

FROM Orders

GROUP BY Customer

Having SUM(OrderPrice)<2000

## HTML5

**Semantic HTML**

- Header/footer elements

- Figure element

- Hgroup element

- Canvas element

**Useful HTML**

- Regex support in input fields

- use <labels>

- No more types for scripts and links

- Built in form validation for email types

- Placeholders for inputs

- LocalStorage API

## What are specific programming challenges you have faced in professional work?

**PHP / Architecture**

* Recursive algorithms for sorting/editing file structure

**Database design / SQL**

* Friend/plan/following database relationship
* SQL query for geo distance

**Uploading assets/videos asynchronously with backend authentication**

* first send GET request to upload\_token (asset controller), then on the back end, cache the token with the user’s ID
* Take the response (with the token)
* Create a hidden input element with the token in it
* POST the new form with the asset, then on the back end, check for the token in the cache

## what is CSS specificity?

- order in which css rules are applied

### what are SOLID principals?

# Open Questions

favorite/least favorite js framework?

what are your design principles?

what is your favorite thing you have worked on?

favorite/least favorite part of javascript?

favorite/least favorite part of jquery?

favorite part of css3?

what is your workflow?

# Javascript Questions

1. why would you bind an event to the parent element?
   1. because with dynamically added content, you don’t have to define another event
   2. also because it’s faster when there are a lot of child elements
2. Write a function in js that sorts an object of names alphabetically
3. Reverse a String
   1. By letter: s = s.split(‘’).reverse().join(‘’);
   2. By word: s = s.split(‘ ‘).reverse().join(‘ ‘);
   3. reverse a string by words (recursively)
4. Implement “Map” in JavaScript (i.e. map([1, 2, 3], func) → [func(1), func(4), func(9)]
   1. Next implement “asyncMap” where “func” is asynchronous

# Programming Questions

1. Write a program that returns true when a string is a palindrome (can be read the same forwards and backwards)
   1. Recursive solution, base case checks size of string, regular case checks first and last letter equivalent and returns smaller version of the word (with the first and last letters cut off)
2. Write a program that returns whether an integer is a power of two
   1. Recursively divide the number by 2 and check whether the modulo 2 of that number is 0. Base case is 1 which means the number was a power of 2.
3. Reverse an array only using a string
   1. For loop through half of the array swapping elements with a temp variable
4. Given a list of elements, return all permutations of those elements
   1. (1, 2, 3) => (1, 2, 3), (1, 3, 2), (3, 1, 2)...etc
   2. recursive solution
5. Given a staggered array of strings (different lengths) (i.e. [[“abc”, “de”, “fgh”], [“ik”, “jlm”], [“put, “ora”, “bon”]]), print all permutations (only going from left to right)
6. Write a function that returns the fib number given # in sequence
   1. now re-write that function so it’s faster by caching results (global array, not localStorage)
7. Write an algorithm in Python that is a concordance. i.e. takes in a long string of text and outputs the # of each word in alphabetical order
8. Write an algorithm that determines the “best match” for a word compared to an array of words. (function takes 2 inputs, a string, and an array of strings)
9. Write a function that given a staggered array (an array of arrays of different lengths), each containing strings of different lengths, returns all permutations of the strings (in order)
10. Write algorithms for all the common set operations (union, join, etc.) on arrays
11. Write an algorithm that determines the outcome of a tic tac toe game

Back end/PHP

Why did you choose to use PHP/MySQL/apache for your web projects?

The criteria we used for selecting our development environment were: **platform independence, speed, power, acceptance level, and free availability**. The Apache/MySQL/PHP combination provides a platform-independent solution that is very fast and provides the power we need to do some complex server-side scripting. The combination is available for free and has reached a high level of acceptance, so much so that Red Hat is distributing it bundled with its Linux 7.0 release. PHP was not appropriate for the scheduler (partly because the scheduler does not necessarily run on the same machine as the web server), so we used Java due to its platform independence and the ease of development that Java provides

**If you go to your website and it is blank/broken, what are the steps you take to debug?**

* Ping server, make sure it is responding
* PHP log

**How would you implement “Remember Me” functionality on a website login page?**

* if someone has remember me selected and they log in successfully, store a cookie with the user name and a token and store the token in the database
* the next time the person goes to the site, see if the cookie token matches the database lookup

## how would you implement sessions/logging in/etc?

**Questions to ask**

* what size teams do you work on?
* how is work delegated within teams?
* what’s your favorite reason for working there?
* what’s the typical coding setup?
* what’s the work environment like?