php, getting started

- Hey folks, you've found the getting started in php short for the web development course at NC State's computer science department. I'm Ben Watson, director of the Visual Experience Lab there.
    - (php, getting started)
    - (web development course)
    - (Ben Watson)
    - (VXLab)
- In this short, we'll talk about php and server side scripting, what it's used for, and how to get started with it.
        - (php…)
    - php is one of many server side scripting languages. It's extremely widespread at this point (about 40% of websites). Others include python, ruby and even js through node.js.
            - (...: a server side scripting language)
        - this means that it runs on servers, not in client browsers (like js)
            - (a server and a browser, with php in the server, js in the browser)
    - server side scripting serves many purposes:
        - It abstracts away many of the differences between browsers. Php can serve html and js that depends on browser
            - (php -> html/js -> browser)
        - It lets you access server resources, e.g. database contents and computer resources. Access directly from browser can be risky.
            - (php -> mysql -> html/js -> browser)
        - It takes compute load from the client
            - (php/server computes, js/client displays)
- You can find more detail on jquery at
    - First, php's online documentation, which is excellent
        - (online documentation)
    - Next, the book php and mysql, novice to ninja, by Yank and published by O'Reilly
        - (yank's novice to ninja)
- By way of history
    - Rasmus Lerdorf wrote php back in 1994.
        - (Rasmus Lerdorf, 1994)
    - He was building dynamic web pages and wanted to have tools. His concerns were developer concerns.
        - (ease development of dynamic pages)
    - It developed organically and by accident. It's a bit messy. No well defined standard exists — they are working on it!
        - (no standard, a bit messy)
- Unlike client side coding, there aren't really any great in browser development environments. The best way to start is to install a php stack on your machine.
        - (install it to begin coding with it)
    - A php stack is the set of software you need to run php
            - (you install a "stack" of software)
        - the typical stack is called w/m/lamp, for windows apache mysql php. The m is for mac, the l is for linux.

        - (wamp: windows, apache, mysql, php)
        - apache is a web server, mysql is a database server. We'll talk about mysql soon.
    - There are many free wamp installs, we like xampp.
        - (one possible wamp install: xampp)
- After you install xampp, you put your web content including php in xampp/htdocs. E.g., test.php.
    - (put your php and web content in xampp/htdocs)
- To get some php into your html, you just use the <?php tag.
        - (getting php in your document: <?php tag
    - For example, <?php echo date('l, F jS Y.'); ?> creates a date string
        - (<?php echo date('l, F jS Y.'); ?>)
    - The php tag is replaced with the resulting tx/html/css/js. In this example, the code above is replaced with today's date:
        - (Sunday, April 1st 2012.)
- To make sure the php interpreter runs, give your code a php rather than a html suffix
    - (use the .php suffix)
- If you want to keep your php and client code separate, you can use the include command
    - (<?php include "mycode.php" ?>)
- And that's it! I hope you found this short useful.