"Modern PHP" (2015)

Language Features

The New PHP
-With additions of features like namespaces, closures, traits, etc., PHP has advanced far in encapsulation and allows moving away from massive monolithic apps.

-For version control, check out Vagrant. For provision tools, look into Ansible, Chef, and Puppet.

-Composer allows for easy dependency management

-PSR,by PHP-FIG community working group has become widely accepted PHP coding standard

-PHPUnit as powerful unit testing tool

-PHP FastCGI process manager

-PHP engine – parses, interprets, and executes PHP code.

-Addition of HipHop Virtual Machine PHP engine, built by Facebook. Exists as alt to older Zend engine. FB also built Hack, white is built on top of core PHP, as a PHP version of Typescript of sorts, with static typing, new data structures, etc.. Both use JIT (Just In Time) compiler for increased speed.

-New features make PHP much more suitable for developing command line tools

Namespaces
-Organize PHP files into a virtual hierarchy similar to standard filesystem directory structures

-Each namespace has it's own global scope. Used in packages to prevent name clashes across files and packages, as well as package uploaders. Vendor + package specific namespaces.

-Namespaces allow import of code from any other package, whether created by another user (ex. *Laravel)* or created in house, providing a sandboxed execution environment, to fully prevent name clashes, as well as organize code, to make easy for re-use, etc..

-Sub-namespaces prevent clashes inside same main namespace by further breaking namespace down

-Declare namespace as first line in PHP file via syntax:
  namespace MyVendor\MyNamespace;

-Can declare nested namespace via:
  namespace MyVendor\MyNamespace\MySubNamespace;

-After deceleration of namespace, all code under this declaration now exists in defined namespace. Any other file that also declares that same namespace also exists in same namespace as other same namespace files.

-A namespace acts as an encapsulation platform for grouping and organizing PHP classes together, just how directories organize and group files.

-Recommended to store PHP files in directory structure matching namespace hierarchy