

CHAPTER II

BASIC OF THEORY

II.1 Programming Language : Briefly Description

Programming language is collection of code program languages to create applications and leads to highly entangled and unmaintainable code. Programming language using logic to build the application and data storage in an application to save the data. [1]

II.2 Definition of PHP Programming Language

PHP programming language is a combine of HTML code with server side programming languages to create dynamic web pages. PHP can be save at hosting and the hosting can be accessed via domain name. Programmer choose PHP because easiness of PHP and it's free. [1]

II.3 History of PHP Programming Language

PHP as it's known today is actually the successor to a product named PHP/FI. Created in 1994 by Rasmus Lerdorf, the very first incarnation of PHP was a simple set of Common Gateway Interface (CGI) binaries written in the C programming language. Originally used for tracking visits to his online resume, he named the suite of scripts "Personal Home Page Tools," more frequently referenced as "PHP Tools". Over time, more functionality was desired, and Rasmus rewrote PHP Tools, producing a much larger and richer implementation. This new model was capable of database interaction and more, providing a framework upon which users could develop simple dynamic web applications such as guestbooks. In June of 1995, Rasmus released the source code for PHP Tools to the public, which allowed developers to use it as they saw fit. This also permitted - and encouraged - users to provide fixes for bugs in the code, and to generally improve upon it. Ough it lived a short development life, it continued to enjoy a growing popularity in still-young world of web development. In 1997 and 1998, PHP/FI had a cult of several thousand users around the world. A Netcraft survey as of May, 1998, indicated that nearly 60,000 domains reported having headers containing "PHP", indicating that the host server did indeed have it installed. This number equated to approximately 1% of all domains on the Internet at the time. Despite these impressive figures, the maturation of PHP/FI was doomed to limitations; while there were several minor contributors, it was still primarily developed by an individual. [4]

II.4 Benefit of PHP Programming Language

There are two benefit using PHP as programming language to build a website. There are :

1. PHP can interact with many different database languages such as MySQL, SQL Server, Mongo DB and others. But a lot of programmer prefer using MySQL as their database.
2. PHP and MySQL are compatible with Apache server. Apache server is a free server. Third, PHP can run on multiplatform such as Linux, Windows, Mac, and Unix Servers.
3. Easy to code because the syntax is easy to use.
4. PHP is open source, so people is free to use and doesn't need to pay.
5. Have a lot of documentation at the internet, so it will help to fix the code when got an error.
6. When deploy on the Apache server, the cost is cheaper than using other languages.
7. A lot of people know PHP and other programmer can help when problems occur. [3]

II.5 Definition of Programming Framework

A *framework* is a hierarchical directory that encapsulates shared resources, such as a dynamic shared library, nib files, image files, localized strings, header files, and reference documentation in a single package. Multiple applications can use all of these resources simultaneously. The system loads them into memory as needed and shares the one copy of the resource among all applications whenever possible.

A framework is also a bundle and its contents can be accessed using Core Foundation Bundle Services or the Cocoa NSBundle class. However, unlike most bundles, a framework bundle does not appear in the Finder as an opaque file. A framework bundle is a standard directory that the user can navigate. This makes it easier for developers to browse the framework contents and view any included documentation and header files.

Frameworks serve the same purpose as static and dynamic shared libraries, that is, they provide a library of routines that can be called by an application to perform a specific task. For example, the Application Kit and Foundation frameworks provide the programmatic interfaces for the Cocoa classes and methods. Frameworks offer the following advantages over static-linked libraries and other types of dynamic shared libraries:

- Frameworks group related, but separate, resources together. This grouping makes it easier to install, uninstall, and locate those resources.

- Frameworks can include a wider variety of resource types than libraries. For example, a framework can include any relevant header files and documentation.
- Multiple versions of a framework can be included in the same bundle. This makes it possible to be backward compatible with older programs.
- Only one copy of a framework's read-only resources reside physically in-memory at any given time, regardless of how many processes are using those resources. This sharing of resources reduces the memory footprint of the system and helps improve performance. [2]

II.6 MVC Pattern

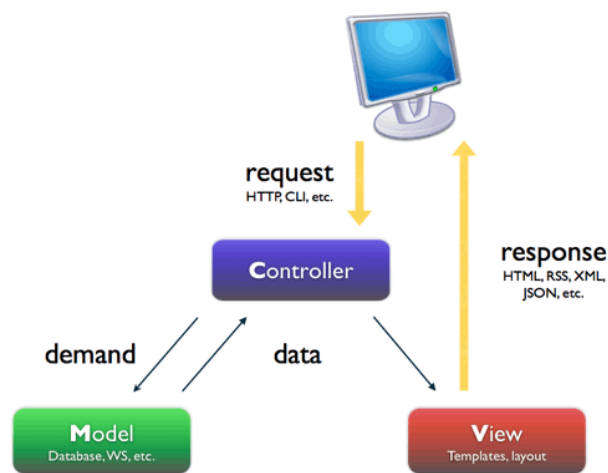


Figure 2.0 MVC Pattern

Source: <http://stackoverflow.com/questions/5966905/which-mvc-diagram-is-correct-web-app>

The MVC design pattern is good for web application because they split the database function, the function code and display separately and structured. With separated folder, the function is neater and doesn't look cluttered rather using manual function. Rather than include the file into the code like PHP Native usually do. [1]