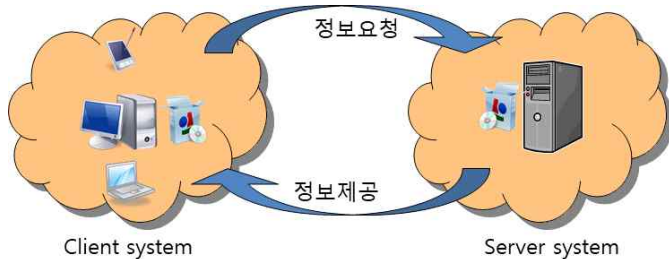


Chapter 01 인터넷, 클라이언트, 서버

1. 클라이언트-서버 환경

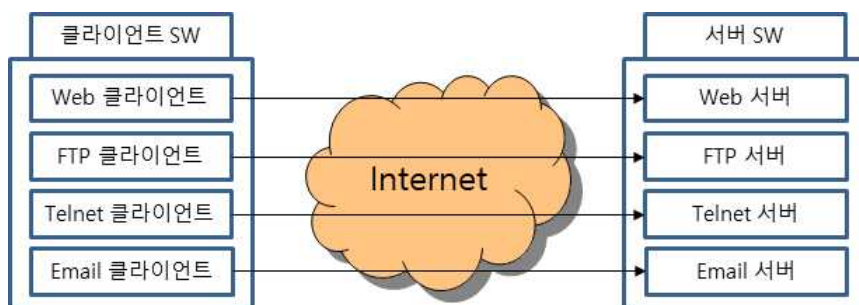


클라이언트-서버는 여러 대의 컴퓨터가 작업을 기능별로 분담하는 환경입니다. 인터넷에 연결되어 있는 모든 컴퓨터들은 클라이언트일 수도 있고 서버일 수도 있고, 둘 다일 수도 있습니다.

그런데 컴퓨터라는 하드웨어들이 동작을 하려면 당연히 소프트웨어가 설치되어 있어야 하는데 이렇게 하드웨어에 소프트웨어가 설치된 것을 시스템이라고 부르겠습니다.

클라이언트는 서버에 정보를 요청하는 시스템이고 서버는 클라이언트의 요청에 의해서 정보를 제공하는 시스템입니다. 즉 인터넷을 통해 클라이언트는 자신이 지정한 서버로부터 정보를 요청할 수 있고, 서버는 많은 정보를 저장하고 있다가 정보를 요청한 클라이언트에게 정보를 보낼 수 있습니다. 여기서 컴퓨터라고 하지 않고 시스템이라고 한 것은 이유가 있습니다. 컴퓨터라고 하면 단순히 하드웨어를 의미하지만, 시스템이라 함은 하드웨어와 소프트웨어를 합친 것으로 이해되기 때문입니다. 클라이언트-서버라고 하는 것은 좁은 의미로는 소프트웨어를 의미하는 것이 맞다고 볼 수 있습니다.

2. 클라이언트-서버 소프트웨어(프로그램)

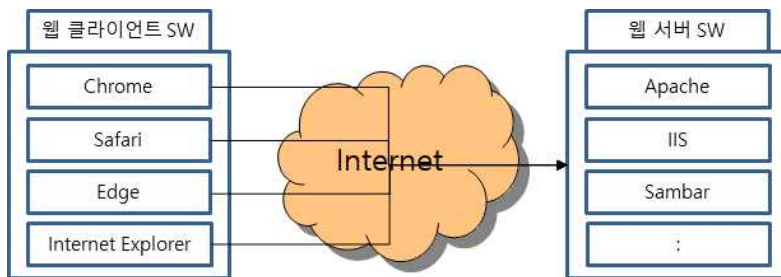


인터넷을 통해 활용할 수 있는 서비스는 여러 가지가 있습니다. 예를 들어 web, ftp, email, telnet 등이 있습니다. 이중 Web 서비스를 일반 사용자 이러한 서비스를 이용하거나 제공하기 위해서 각 서비스에 맞는 클라이언트용 소프트웨어를 사용해야 하며, 서버용 소프트웨어도 설치하여 동작시켜야 합니다. 따라서 각 서비스에 사용하는 클라이언트용, 서버용 소프트웨어가 별도 있습니다. 서버용 소프트웨어들은 시스템 관리자들이 사용하는 것이기 때문에 일반 사용자들은 알 필요는 없습니다.

인터넷 서비스	클라이언트 소프트웨어	서버 소프트웨어
웹(world wide web)	Chrome, Edge, Safari, Internet Explorer	Apache, IIS(Windows), Sambar,
e-mail	Outlook, Eudora, MailTrim	MDaemon, IceWarp, FTgate
FTP (File Transfer Protocol)	FileZilla Client FlingFTP, ClassicFTP	IIS(Windows), serv-u, War FTP, FileZilla Server, WS FTP
Telnet	Pinetern, Wintern, Netterm, telnet 클라이언트(win 10),	telnet 서버(win 10) telnet-server(Linux)

클라이언트 소프트웨어 사용자들은 서버 컴퓨터에 어떤 소프트웨어가 설치되었는지는 몰라도 됩니다. 사용자들은 원하는 서비스에 적합한 소프트웨어를 이용해 서버에 접속하기만 하면 됩니다.

3. 웹클라이언트-웹서버

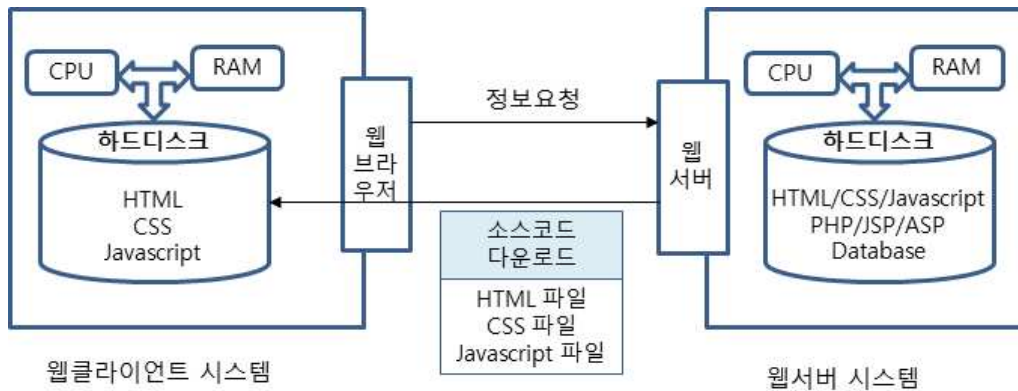


클라이언트 컴퓨터들이 인터넷에 연결되어 있다고 해도 웹 접속용 소프트웨어(chrome, edge, safari 등)가 설치되지 않으면 웹에 접속할 수 없게 됩니다. 또한 웹 접속용 소프트웨어로 접속을 시도해도 지정한 서버 컴퓨터에 정보를 제공할 수 있는 서버 소프트웨어가 실행되어 있지 않다면 클라이언트에 정보를 제공할 수 없게 됩니다. 웹클라이언트 소프트웨어를 일반적으로 웹브라우저라고 부릅니다. 웹클라이언트와 웹서버 관련 소프트웨어들은 위의 표를 참조하세요.

이렇게 웹 관련 소프트웨어가 클라이언트/서버에 설치되면 사용자들은 웹서비스를 이용할 수 있게 됩니다. 위 그림을 보면 사용자는 웹브라우저의 주소창에 자신이 원하는 사이트 예를 들어 www.naver.com 같은 사이트 주소를 입력하여 접속되면 원하는 정보를 웹브라우저 화면으로 볼 수 있습니다.

4. 웹 개발 기술, 웹 프로그래밍

웹 개발자 즉 웹 프로그래머가 되기 위해서는 위에서 설명한 클라이언트 기술 분야와 서버 기술 분야를 구분할 줄 알아야 합니다. 클라이언트 분야 개발자와 서버 분야 개발자의 역할이 분리되어 있는 것은 아니지만, 각각의 기술을 구분하여 이해하는 것은 매우 중요합니다. 특히 웹 개발자로 처음 시작하는 사람들에게 웹서비스 시스템의 동작환경을 이해하는데 많은 도움을 줍니다. 단순히 개발 기술들이 어떻게 있다가 아니라, 각 개발 기술들이 서로 어떻게 동작하고 또 시스템 내에서는 어떤 역할을 하는지에 대한 이해가 중요합니다.



위 <그림>에서 사용자는 웹 브라우저를 통해 웹 서버에 접속해서 정보를 가져옵니다. 이때 HTML, CSS, Javascript 파일들은 클라이언트 시스템의 하드디스크에 다운로드 되고 CPU, Memory 장치들을 이용하여 웹브라우저에서 실행됩니다. 이렇게 하는 이유는 모든 파일들을 웹서버에서 실행하게 하면 웹서버에 너무 많은 작업량이 몰리기 때문에 부하가 걸릴 수 있기 때문입니다.

이런 방식으로 동작되는 프로그램들을 Client Side Include 라고 하며 이러한 기술 분야를 Front-end 기술이라고도 합니다.

반면에 PHP, JSP, ASP 등의 파일들은 서버 시스템의 CPU, Memory 장치들을 이용하여 실행되고 그 결과가 웹 브라우저에 전송됩니다. 이런 방식으로 동작되는 프로그램들을 Server Side Include 라고 하며 이러한 기술 분야를 Back-end 기술이라고도 합니다.

이것이 Front-end(Client Side Include)와 Back-end(Server Side Include)를 구분하는 기준이라고 보면 됩니다.

정리하면, 웹 서비스와 관련된 모든 파일들은 웹서버 시스템에 저장되어 있지만 Client Side Include 관련 파일들은 웹클라이언트 시스템에 다운로드되어 실행되고, Server Side Include 관련 파일들은 웹서버 시스템에서 실행된다는 겁니다.

비록 동작환경의 이해를 위해 Front-end(Client Side Include)와 Back-end(Server Side Include)를 구분하고는 있지만 웹 개발자에게는 두가지 모두를 수행해야 하기 때문에 큰 의미는 없습니다. 즉 대부분의 웹 개발자들은 Front-end와 Back-end 모두를 담당하게 됩니다.

다만 학습순서는 Front-end(Client Side Include) 관련 기술들을 먼저 학습해야 합니다.

본 도서에서는 Back-end에 해당되는 Server Side Include 관련 내용을 다루고 있습니다. Front-end 관련 내용을 접하지 않은 독자들은 이 부분을 먼저 학습하기 바랍니다.

4.1 웹 클라이언트 프로그래밍, Front-end, Client Side Include

웹 클라이언트 개발은 Front-end 또는 Client Side Include 개발이라고 표현하기도 합니다. 주로 웹 브라우저에 정보를 보여주기 위해서 사용되는 기술들로 HTML(Hyper Text Markup Language), CSS(Cascading Style Sheet), Javascript 기술들을 말합니다.

HTML은 웹 정보 또는 콘텐츠들을 잘 보여지게 위해 구조를 설정하는 기술입니다. CSS는 웹 브라우저에 정보 또는 콘텐츠들의 배치와 일관된 디자인과 관리를 하기위한 기술입니다. HTML, CSS 기술은 약간은 디자인과 관련된 분야이긴 하지만 웹 개발자들은 당연히 알고 있어야 합니다. 아무리 좋은 정보들도 시각적으로 어색하게 표현되거나 배치되어 있으면 사용자들로부터 외면받기 때문이죠. 물론 웹 그래픽이라는 디자인 분야가 별도로 있기는

하지만 이건 또 다른 직업군입니다.

Javascript는 프로그래밍 언어입니다. Javascript는 웹 브라우저에 보이는 정보들을 사용자의 반응에 따라 동적으로 보여지도록 하기 위해 사용되는 기술이며 웹 개발자라면 누구나 알고 있어야 하는 중요한 기술입니다. Front-end 개발, Client Side Include 개발이라고 하는 것의 핵심이 바로 Javascript 프로그래밍 언어를 통한 개발이라고 할 수 있습니다. 물론 Front-end 분야에 더 많은 설명이 필요합니다만, 다른 자료들을 참고하기 바랍니다.

Client Side Include 용어에 대해 좀 살펴보겠습니다. Include 대신 Program, Script라는 용어로 사용하기도 하며 이들 용어에서도 느껴지듯이 프로그래밍 코드입니다. 한마디로 웹 클라이언트 시스템에서 실행되는 프로그램이라는 의미이며, Front-end 프로그램과 같은 의미로 사용됩니다. 여기에서 사용되는 프로그래밍 언어는 앞서 언급한 Javascript 언어외에도 VBScript 언어도 있습니다만, Javascript 언어가 압도적으로 많이 사용됩니다.

Javascript 언어와 관련된 Client Side Include에 대해 좀 더 자세히 살펴보겠습니다. 앞서 설명한 대로 Client Side Include는 웹 브라우저에 정보를 보여주기 위한 것이라고 했는데 어떤 것들이 있는지 알아 보겠습니다.

우선 Client Side Include는 클라이언트 시스템에서 처리되어 웹브라우저에서 실행됩니다. 다운로드되기 때문에 모든 소스코드들은 공개 됩니다. Client Side Include 파일들은 클라이언트 시스템에서 실행되기 때문에 혼자서 실행할 경우에는 웹서버에 접속하지 않고도 실행이 가능합니다. 즉 HTML, CSS, Javascript 학습은 웹서버뿐만 아니라 인터넷에 연결되어 있지 않아도 가능합니다.

웹 사이트에 접속하다 보면 Popup 윈도우 생성, 메뉴구성, 화면전환효과, 버튼롤오버 등의 기능을 보게 되는데 이런 것들을 Javascript 언어로 프로그래밍하는 겁니다.

다시 한번 강조하지만 Client Side Include는 웹클라이언트 시스템에서 실행되지만, 소스코드 자체는 웹서버 시스템에 저장되어 있다는 점을 명심하세요.

4.2 웹 서버 프로그래밍, Back-end, Server Side Include

일반적으로 웹 개발자라고 하면 웹 서버 프로그래밍 개발자를 말하며 Back-end 개발자라고도 합니다. Back-end 프로그래밍에서 개발된 파일 또는 소스코드를 Server Side Include라고 하며, Include 대신 Program, Script 라는 용어를 사용하기도 합니다. 웹 서버 시스템에서 실행되는 프로그램이라는 의미이며, 주로 데이터베이스와 연관된 프로그램입니다.

개발에 사용되는 대표적인 프로그래밍 언어로는 PHP(Professional Hypertext Preprocessor), JSP(Java Server Page), ASP(Active Server Page) 등이 있으며, 본 도서에서는 PHP 언어에 대해 학습합니다.

Server Side Include(Back-end) 파일들은 클라이언트에 소스코드가 전송되지 않기 때문에 보안유지 가능하며 주요 역할은 다음과 같습니다.

- 데이터베이스의 데이터 삽입, 수정, 검색
- 게시판, 회원가입/인증
- 정보보안 기능

Server Side Include 파일들은 웹서버가 동작하는 환경에서 실행되기 때문에 혼자서 실행할 경우에는 인터넷이 연결되어 있어야 합니다.

Back-end 분야에 관련된 기술들은 매우 다양하며 본 도서에서 다루는 PHP 언어는 가장 기본적이며, Back-end 분야의 일부에 불과하기 때문에 보다 자세한 내용은 다른 자료를 참

고하기 바랍니다.

4.3 웹서버

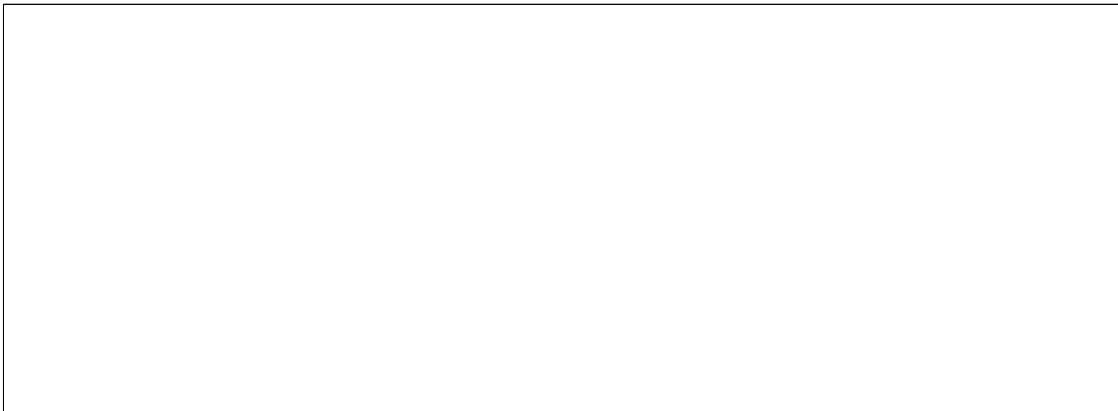
웹서버 시스템은 컴퓨터(CPU, Memory, Harddisk)와 웹서버 소프트웨어로 구성됩니다. 컴퓨터만으로는 웹서비스를 할 수 없으며, 웹서버라는 소프트웨어가 설치되어 있어야 웹서비스가 가능합니다. 웹서버를 하드웨어 즉 컴퓨터를 지칭하기도 하지만 엄밀하게 말하면 소프트웨어라고 볼 수 있습니다. 즉 컴퓨터 전원이 켜져 있어도 웹서버 소프트웨어가 실행되지 않게 되면 접속이 불가능합니다.

웹서버 소프트웨어에는 Apache, IIS(Internet Information Server), Sambar 등이 있는데 Apache가 가장 많이 사용됩니다. 본 도서에서도 Apache 웹서버를 이용하여 학습합니다.

웹서버는 클라이언트 시스템의 웹브라우저가 웹사이트에 접속 또는 정보를 요청할 때 반드시 통과해야 할 대문과 같은 역할을 합니다. 웹서버 관리자는 웹서버 소프트웨어를 상황에 맞게 설정하여 웹사이트를 관리합니다.

웹서버 관리는 프로그래밍 분야와 밀접한 관련은 없지만 Back-end 분야라고 볼 수 있습니다. 웹서버 관리에 대한 이해가 되어야만 프로그램 개발시에 적절한 대응을 할 수 있기 때문입니다.

4.4 데이터베이스 관리시스템(DBMS : Data Base Management System)



Chapter 02 개발환경 구축

웹서버 프로그램을 개발하기 위해서는 관련 소프트웨어들이 컴퓨터에 설치되어야 하는데 본 도서에서는 프로그래밍 언어는 PHP, 웹서버는 Apache, DBMS는 MariaDB(MySQL)을 이용합니다. 이것들은 소프트웨어이며 개발도구라고 불리기도 합니다. 개발도구들은 각각 별도로 설치할 수도 있지만 상호 연결을 설정하는 과정들이 복잡하기 때문에 이들을 한번에 설치할 수 있는 통합 개발환경(패키지)을 이용하여 구축합니다. 여기에 제시하고 있는 소프트웨어들은 모두 무료로 사용할 수 있는 오픈소스입니다.

우선 PHP, Apache, MariaDB 공식 홈페이지를 방문하여 각 개발도구들의 최근 동향을 살펴봅시다.

1. PHP

PHP 프로그래밍 언어의 공식 홈페이지는 『<https://www.php.net>』입니다. 접속해보면 다음의 화면이 나오는데 PHP 엔진 다운로드뿐만 아니라 PHP 학습 매뉴얼도 제공하고 있습니다.



우선 Downloads 메뉴를 클릭하면 다음과 같이 다운로드 가능한 목록이 제시됩니다. 현재 안정된 최신 버전은 8.1.7 이고, Windows 설치용은 맨 아래에 있으며 이것을 클릭하면 다음의 압축파일이 다운로드 됩니다. 본 도서에서는 통합 패키지를 설치하기 때문에 PHP 엔진의 개별적인 설치하지 않습니다.

php-8.1.7.tar.gz

Current Stable PHP 8.1.7 ([Changelog](#)) ¶

- [php-8.1.7.tar.gz](#) (sig) [19,252Kb] 09 Jun 2022
sha256: 5f0b422a117633c86d48d028934b8dc078309d4247e7565ea34b2686189abdd8
- [php-8.1.7.tar.bz2](#) (sig) [15,101Kb] 09 Jun 2022
sha256: b816753eb005511e695d90945c27093c3236cc73db1262656d9fadd73ead7e9d
- [php-8.1.7.tar.xz](#) (sig) [11,444Kb] 09 Jun 2022
sha256: f042322f1b5a9f7c2dec84b7086ef676896c2f7178739b9672afafa964ed0e5
- [Windows downloads](#)

매뉴얼은 Documentation 메뉴 → English 링크를 클릭하면 영문 PHP 사용법에 대한 설명이 나옵니다. PHP 문법에 대한 모든 내용을 수록하고 있습니다.

Language Reference ¶

- [Basic syntax](#)
 - [PHP tags](#)
 - [Escaping from HTML](#)
 - [Instruction separation](#)
 - [Comments](#)
- [Types](#)
 - [Introduction](#)
 - [Booleans](#)
 - [Integers](#)
 - [Floating point numbers](#)

2. Apache

Apache 웹서버의 공식 홈페이지는 『<https://apache.org>』입니다. 접속해보면 다음의 화면이 나옵니다.



상단의 Downloads 메뉴를 클릭하면 Apache 웹서버를 다운로드할 수 있지만 여러번 링크를 클릭해야 해서 다음의 URL을 통해 직접 접속해서 다운로드할 수 있습니다.

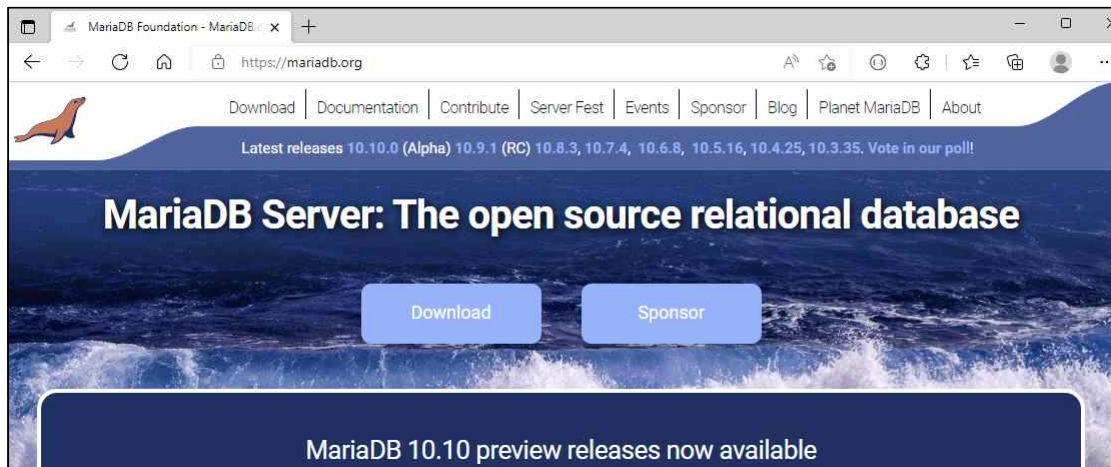
<https://www.apachehaus.com/cgi-bin/download.plx>

위 URL 페이지에서 Apache 2.4.54이 최신 버전으로 등록되어 있으며 이것을 클릭하면 다음의 파일이 다운로드 됩니다. 본 도서에서는 통합 패키지를 설치하기 때문에 Apache 웹서버의 개별적인 설치하지 않습니다.

[httpd-2.4.54-ol11lp-x86-vs17.zip](#)

3. MariaDB

MariaDB DBMS의 공식 홈페이지는 『<https://mariadb.org>』입니다. 접속해보면 다음의 화면이 나오는데 MariaDB 소프트웨어 다운로드뿐만 아니라 학습 매뉴얼도 제공하고 있습니다.



우선 Download 버튼을 클릭하면 다음과 같이 다운로드를 위한 선택목록이 제시됩니다. 사용자 환경에 적합한 항목을 선택하여 다운로드 할 수 있습니다.

MariaDB Server	MariaDB Repositories	Connectors
<p>This is a short term release with support lasting one year only. For production use we suggest the 10.6 release series.</p>		
<p>MariaDB Server Version</p> <p>MariaDB Server 10.8.3</p>		
<p>Display older releases: <input type="checkbox"/></p>		
<p>Operating System</p> <p>Windows</p>		
<p>Architecture</p> <p>x86_64</p>		
<p>Package Type</p> <p>MSI Package</p>		

다음의 파일이 다운로드되는데 안정적인 최신 버전은 10.8.3입니다. 본 도서에서는 통합 패키지를 설치하기 때문에 MariaDB의 개별적인 설치에 하지 않습니다.

mariadb-10.8.3-winx64.msi

매뉴얼은 상단의 Documentation 메뉴를 클릭하면 PDF 파일로 다운로드할 수 있는 페이지가 나옵니다.

MariaDB vs MySQL

MySQL 핵심개발자들이 MySQL에 대한 오라클의 라이선스 정책이 자신들과 맞지 않는다는 이유로 오라클사를 탈퇴해서 만든것이 MariaDB 입니다.

MariaDB는 MySQL의 소스코드를 그대로 사용하여 개발했기 때문에 모든 기능이 동일합니다. 차이점은 라이선스 정책인데 MySQL은 소유권이 오라클에 있어서 기업에서 상업적으로 사용하는데 비용이 발생하지만, MariaDB는 오라클사와 무관하기 때문에 무료 사용이 가능합니다.(엔터프라이즈 버전은 유료지만 무료의 범위가 넓습니다.)

4. 통합 패키지

지금까지 개발도구들의 각 공식 홈페이지에서 설치파일을 다운로드하는 과정을 살펴봤습니다. 앞에서 설명했듯이 이들 개발도구들을 개별적으로 설치하면 상호 연결시키는 과정이 필요하기 때문에 설명이 복잡할 수 있습니다. 따라서 이들 3가지 개발도구들을 통합하여 한번에 설치하는 과정을 살펴보겠습니다. 이를 통합 패키지라고 하는데 여러 종류가 있습니다. 대표적인 통합 패키지는 XAMPP, LAMP, MAMP, WAMP 등이 있으며 이것을 서버 스택이라고도 합니다.

4.1 XAMPP

XAMPP는 X(Cross) Apache MariaDB Php Perl 용어들의 약어입니다. X(Cross)는 크로스 플랫폼을 의미하는 것으로 아키텍처가 동일한 컴퓨터(예를 들어 x86 아키텍처)에 설치된 다양한 운영체제(Windows, Linux, OS X)에서 문제없이 동작되는 프로그램 또는 소프트웨어를 말합니다. 따라서 XAMPP는 모든 운영체제에서 동작되는 통합 패키지입니다. 나머지 약어는 아파치 웹 서버, MariaDB DBMS, PHP 언어, Perl 언어입니다.

XAMPP 통합 패키지는 가장 많이 사용되고 있는 PHP 프로그램 개발환경이며, 여기에는 Apache, MariaDB(MySQL), PHP 뿐만 아니라 Control Panel(서버 통합관리 도구), phpMyAdmin(웹기반 DB 관리 도구) 등도 포함되어 있어 웹 개발에 편리성을 제공하고 있습니다.

4.2 LAMP/MAMP/WAMP

공통적으로 사용되는 AMP는 XAMPP와 마찬가지로 Apache MySQL Php 용어들의 약어입니다. L, M, W 등의 영문 첫글자만 다릅니다.

통합 패키지	서버 스택	운영체제
LAMP	Linux Apache MySQL Php	Linux 운영체제에서 동작
MAMP	Mac Apache MySQL Php	Mac 운영체제에서 동작
WAMP	Windows Apache MySQL Php	Windows 운영체제에서 동작

5. XAMPP 통합 패키지 설치

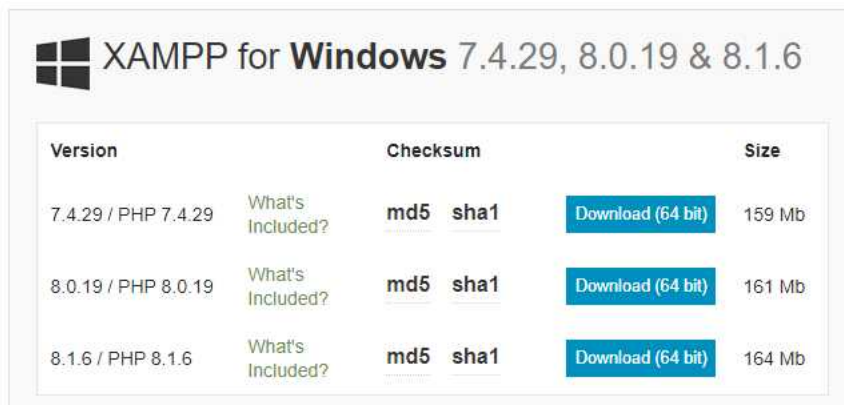
본 도서에서는 웹 개발환경 구성을 위한 XAMPP 통합 패키지를 이용합니다. 설치된 개발도구들을 통해 Apache 웹서버, PHP 언어, MariaDB DBMS 등의 학습을 진행합니다.

XAMPP 통합 패키지의 공식 홈페이지는 『<https://www.apachefriends.org>』입니다. 접속해

보면 다음과 같은 메인 페이지가 표시됩니다.



Download 메뉴를 클릭하면 각 운영체제별 그리고 통합 패키지 버전별로 다운로드 가능한 목록이 표시됩니다. 사용자에게 적합한 운영체제와 패키지 버전을 클릭하면 됩니다.



XAMPP 통합 패키지는 크로스 플랫폼이기 때문에 Windows, Linux, OS X 등의 운영체제를 지원하고 있습니다. 여기서는 Windows의 최신 버전인 8.1.6을 다운로드 합니다.

다운로드를 하면 다음과 같이 파일이 저장되는데, 파일이름에 버전 숫자 8.1.6이 표시되어 있습니다.

xampp-windows-x64-8.1.6-0-VS16-installer.exe

위 다운로드 화면에서 『What's Included?』 링크를 클릭하면 통합 패키지의 각 버전에 포함된 개발도구들의 버전을 알 수 있는데 주요 개발도구들의 버전은 다음과 같습니다.

통합 패키지 버전	개발도구별 버전
7.4.29	Apache 2.4.53, MariaDB 10.4.24, PHP 7.4.29, phpMyAdmin 5.2.0
8.0.19	Apache 2.4.53, MariaDB 10.4.24, PHP 8.0.19, phpMyAdmin 5.2.0
8.1.6	Apache 2.4.53, MariaDB 10.4.24, PHP 8.1.6, phpMyAdmin 5.2.0

다른 개발도구들의 버전은 동일한데 PHP 버전만 다릅니다. 즉 XAMPP 통합 패키지의 버전 숫자는 PHP 버전입니다. 앞서 언급했듯이 XAMPP 통합 패키지는 가장 많이 사용되고 있는 PHP 프로그램 개발환경이기 때문에 PHP 버전을 패키지의 버전으로 표시하고 있습니다.

각 개발도구 공식 홈페이지에서의 버전들은 PHP 8.1.7, Apache 2.5.54, MariaDB 10.8.3 이었는데 통합 패키지에 들어있는 것들 보다는 최신입니다. 최신 버전들을 모아서 통합 패키지 파일을 만들기 위해선 아무래도 시간이 좀 걸리겠죠.

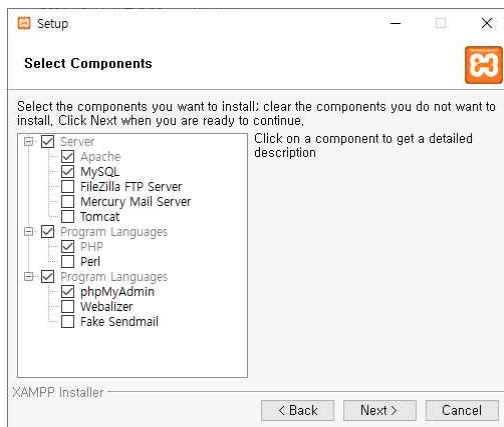
다운로드된 파일인 『xampp-windows-x64-8.1.6-0-VS16-installer.exe』을 실행하여 사용자의 컴퓨터에 설치가 진행됩니다.. 기본적으로 Yes, Next 버튼만을 클릭하면 설치는 됩니다.

5.1 XAMPP 설치

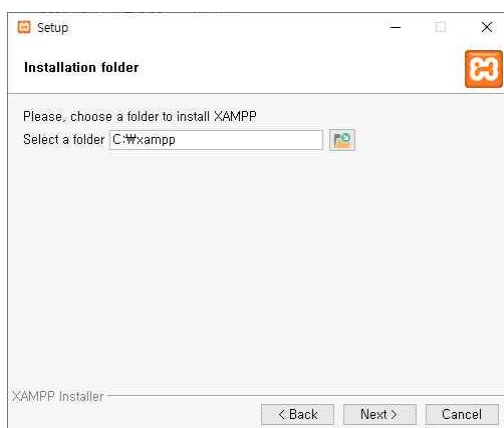
① 설치 초기화면에서 [Next] 버튼을 클릭합니다.



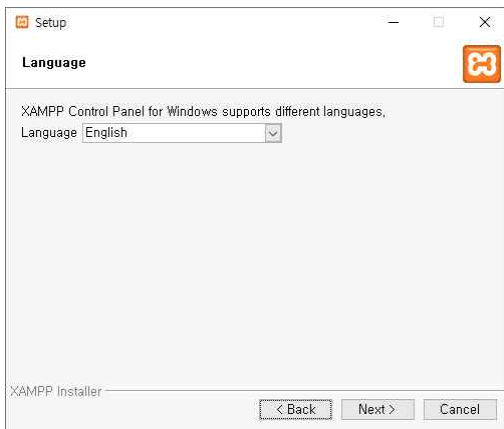
② 설치할 개발도구(컴포넌트)를 선택하는 화면에서 필요한 항목만 체크합니다.



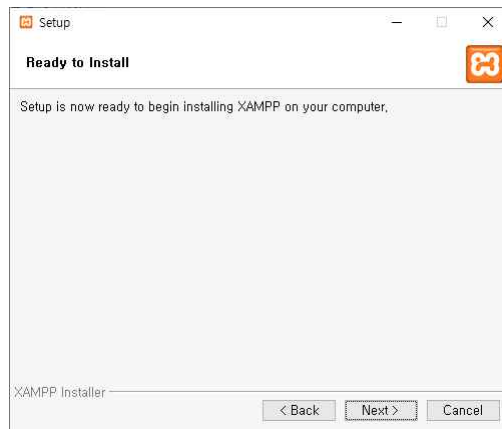
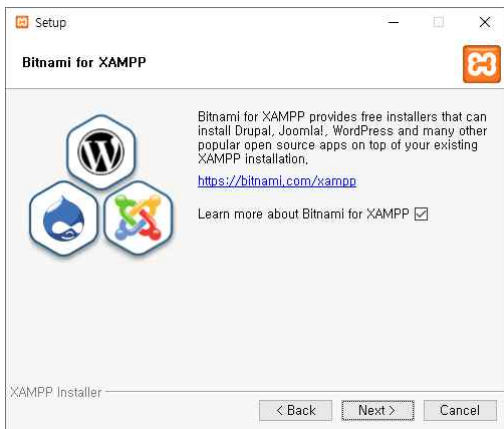
③ 설치 경로를 지정할 수 있는데, 그대로 유지해도 되고 사용자 컴퓨터 환경에 따라 수정해도 됩니다.



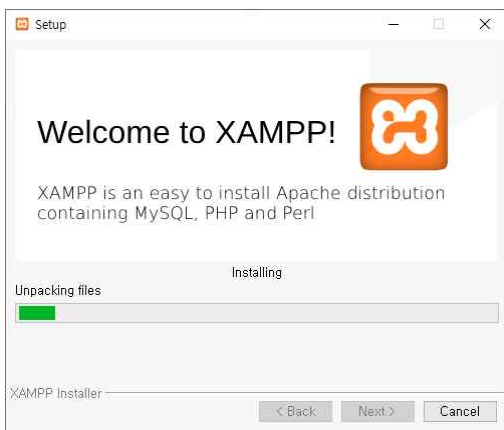
④ 언어 선택 화면에서 English 항목 그대로 유지하고 [Next] 버튼을 클릭합니다.



⑤ 간단한 소개 화면이 표시되고 [Next] 버튼을 클릭후 『Ready to Install』 화면에 다시 [Next] 버튼을 클릭하면 드디어 설치가 시작됩니다.



⑥ 설치과정이 진행됩니다.

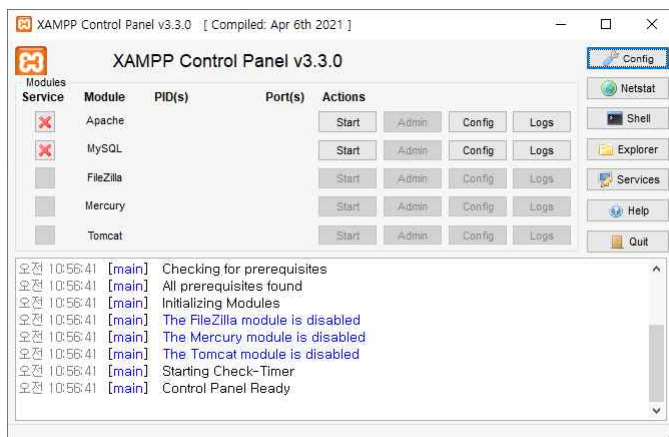


⑦ 설치 완료 직전에 [Windows 보안 경고] 윈도우가 표시될 수 있는데, 하단의 [엑세스 허용] 버튼을 클릭합니다.

⑧ 설치 완료 화면입니다. [Finish] 버튼을 클릭하면 설치 화면 사라집니다.

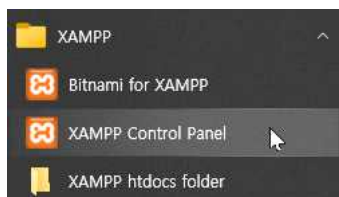


⑨ 위 화면에서 사라지면서 [XAMPP Control Panel] 윈도우가 나타납니다. 이 윈도우의 세부 항목들은 사용자마다 다를 수 있지만 설정을 변경하면 됩니다.



5.2 XAMPP Control Panel 활용

XAMPP Control Panel은 윈도우 시작 메뉴에서 XAMPP 그룹에서 선택할 수도 있고, XAMPP 패키지가 설치된 폴더에서 xampp-control.exe 파일을 실행하여 동일한 윈도우를 동작시킬 수 있습니다. 버전은 3.3.0입니다.



XAMPP Control Panel은 Apache, PHP, MariaDB 등의 개발도구들을 편리하게 관리하고 설정할 수 있도록 구성된 프로그램입니다.

1) Apache 웹서버 시작/중지

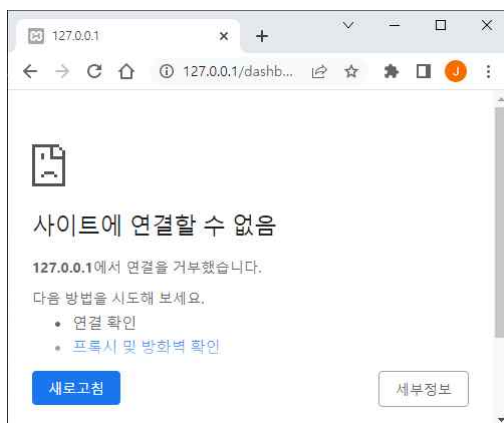
XAMPP Control Panel을 실행하여 Apache 모듈의 [Start] 버튼을 클릭하면 [Stop]으로 변경되면서 웹서버가 시작됩니다.



Apache 웹서버가 정상적으로 동작하는지 확인해 봅시다. 우선 웹브라우저 주소창에 127.0.0.1 또는 localhost를 입력해 봅니다. 다음과 같이 사이트에 연결되면 웹서버가 정상적으로 동작하는 겁니다.



이번에는 [Stop] 버튼을 클릭해서 웹서버를 중지시키고 다시 웹브라우저를 새로 고침하거나 주소창에 127.0.0.1 또는 localhost를 입력해서 접속해 봅니다. 다음과 같이 『사이트에 연결할 수 없음』으로 표시됩니다. 웹서버가 중지된 겁니다. 이렇게 손쉽게 웹서버를 시작하고 중지시킬 수 있습니다. MySQL 서버도 같은 방법으로 시작과 중지를 할 수 있습니다.



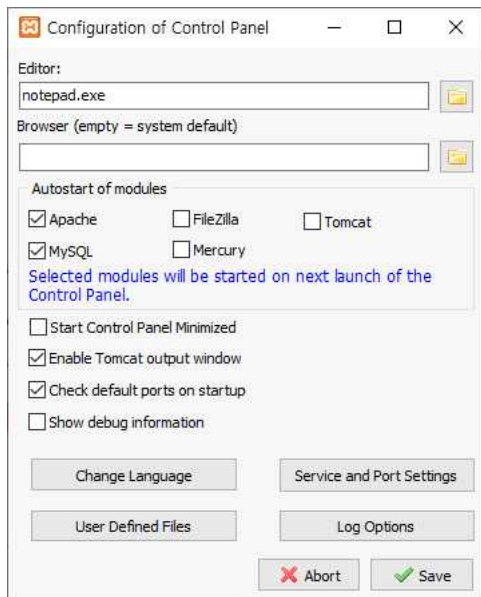
Apache 웹서버를 보다 세부적으로 관리하고 설정하는 방법은 Chapter 03에서 살펴 보겠습니다.

2) Apache/MySQL 자동실행

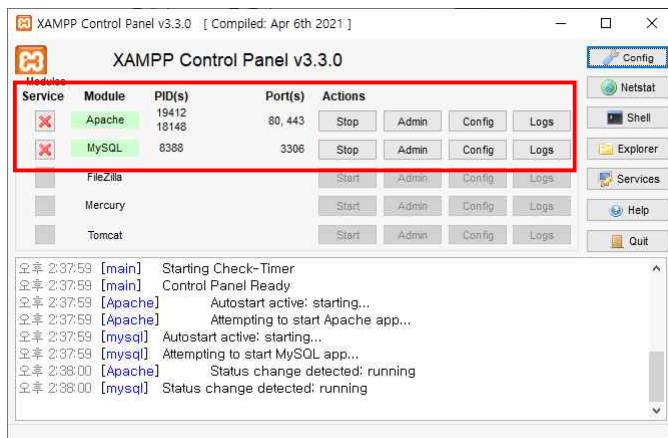
Apache/MySQL 서버의 실행은 Control Panel의 [Start] 버튼을 클릭하여 실행됩니다. 이런 방법이 불편할 경우 자동으로 실행하는 방법이 있습니다. 2가지 방법이 있는데 용도가 다릅니다.

① [Config] 버튼을 이용하는 방법

Control Panel의 우측상단에 [Config] 버튼을 클릭하면 다음과 같은 윈도우 표시됩니다. 여기에서 Apache, MySQL 항목을 체크하고 [Save] 버튼을 클릭하면 됩니다.

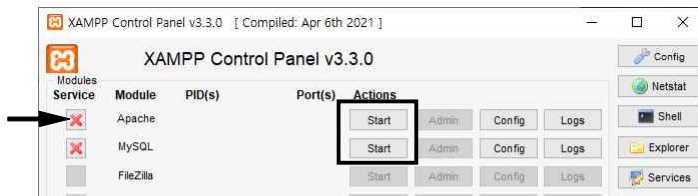


다음으로 Control Panel을 종료하고 다시 실행하면 체크된 항목들의 서비스가 실행됩니다. 반드시 Control Panel을 재실행해야 합니다. 아래 윈도우처럼 Apache와 MySQL이 실행되는 것을 알 수 있습니다.

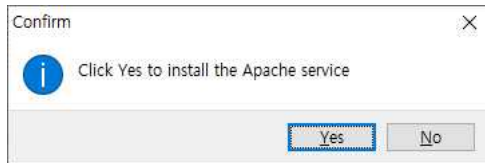


② Service 항목을 이용하는 방법

이 방법은 컴퓨터가 부팅될 때 Apache, MySQL 모듈들을 자동으로 실행하는 방법입니다. Control Panel의 왼쪽에 서비스 항목 체크하면 됩니다. 단 자동으로 실행하고자 하는 서버들은 반드시 중지되어 있어야 합니다.



빨간 [X] 버튼을 클릭하면 다음과 같은 Confirm 윈도우가 표시되고 [Yes] 버튼을 클릭하면 녹색 [✓] 버튼으로 변경됩니다. 이렇게 설정되면 컴퓨터가 부팅될 때마다 선택된 서비스들이 자동으로 실행됩니다.



주의할 점은 자동실행된 서비스들이 계속 동작하게되면 CPU, Memory를 사용하기 때문에 컴퓨터 성능에 영향을 줄 수도 있고, 외부 접속이 허용될 수 있습니다. 따라서 일부러 중지해야 하는 불편함이 있기 때문에 항상 서비스를 운영하는 컴퓨터가 아니라면 설정하지 않는 것이 좋습니다.

XAMPP Control Panel 사용법에 대한 세부적인 내용은 Apache, PHP, MySQL 파트에서 설명합니다.

Chapter 03 Apache 웹서버 관리와 설정

XAMPP Control Panel을 이용하여 Apache 웹서버를 편리하게 관리하고 설정할 수 있습니다.

1. Apache 웹서버 구성 파일 : httpd.conf

구성 파일은 시스템 관리나 인터넷 서비스 등에 필요한 구성을 설정해 놓는 일종의 환경 설정 파일입니다.

여기서 잠시 httpd 파일 이름 뒤에 있는 『d』라는 철자에 대해 알아 보겠습니다. 『d』는 daemon의 약자로 사전적으로는 악령 또는 그리스 신화에서 초자연적인 존재를 의미한다고 합니다. IT분야에서는 사용자가 직접 제어하지 않고, 백그라운드에서 돌면서 여러 작업을 하는 프로그램을 의미합니다. 같은 의미로 상주 프로그램(resident program)이란 단어로 사용되기도 합니다. 나중에 살펴볼 MySQL도 mysqld라는 이름으로 백그라운드 작업 목록에 등록되어 수행합니다.

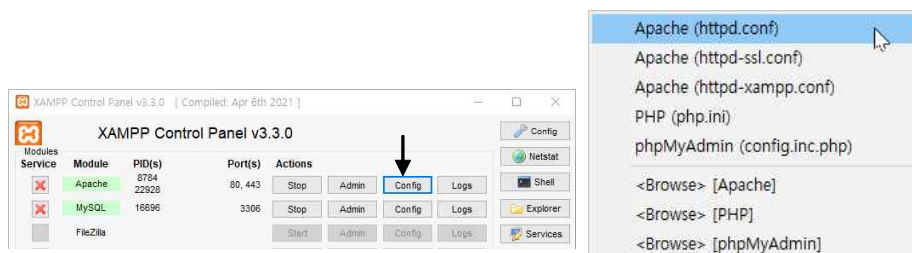
Apache 웹서버는 httpd.conf 파일을 이용하여 여러 가지 환경을 미리 설정해 놓고 있습니다. 흔히 기본값이라고 불리는 형태로 말이죠. 이들 기본값은 웹사이트의 운영 환경에 맞게 변경 즉 재설정할 수 있습니다. 어쩌면 재설정하는 것이 당연합니다. XAMPP의 기본 설정은 공개된 정보이기 때문에 외부에서의 해킹 등으로 문제가 될 수 있기 때문입니다. 웹서버 관리자는 반드시 재설정해야 합니다.

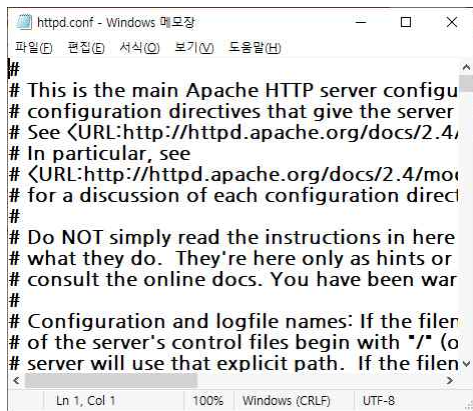
단, 이러한 재설정들은 웹사이트 운영에 영향을 미치기 때문에 정말 조심해서 다뤄야 합니다. 본 도서의 독자들은 학습하는 단계라서 큰 문제가 되지 않고, 실령 문제가 생기더라도 다시 설치하면 되지만 현업 실무에서는 정말 주의해야 합니다. 어쨌든 여러분들도 반드시 httpd.conf 파일 원본은 복사해 놓으세요.

우선 이러한 재설정은 구성 파일을 수정해야 하는데, 이 구성 파일의 이름은 httpd.conf이며, 다음의 경로에 저장되어 있습니다.

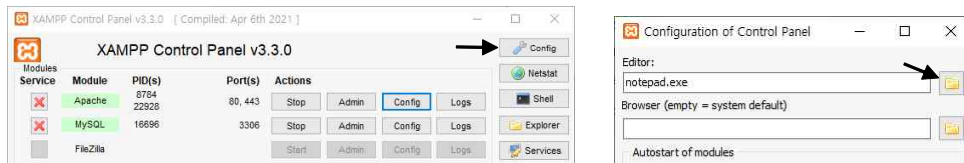
C:\xampp\apache\conf

httpd.conf 파일은 Control Panel을 통해서도 쉽게 불러올 수 있습니다. [Config] 버튼을 클릭하여 나오는 화면에서 Apache(httpd.conf)를 선택하면 미리 설정된 편집기(메모장)를 통해 httpd.conf 파일을 불러옵니다.

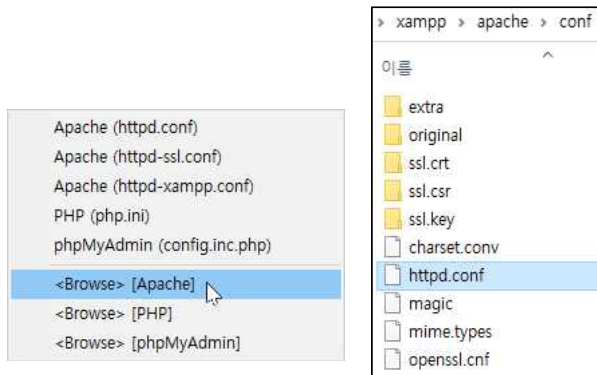




미리 설정된 편집기는 Control Panel의 [Config] 버튼을 통해 변경할 수 있습니다.



또한 아래 화면처럼 <Browser> [Apache]를 선택하면 윈도우 탐색기가 실행되어 httpd.conf 파일이 있는 경로가 나타납니다. 물론 C:\xampp\apache\conf 경로로 연결되지요.



자 그럼 여러분 스스로 관리자라고 가정하고 재설정하는 방법에 대해 살펴봅시다. 여기에서 사용되는 httpd.conf 소스코드의 라인 번호는 XAMPP 통합 패키지의 Windows 최신 버전인 8.1.6 기준입니다. httpd.conf 파일을 열어보면 총 566 라인으로 구성되어 있습니다. # 표시는 주석이며 # 표시 없는 라인을 재설정 합니다.

그리고 httpd.conf 파일이 변경되면 당연히 저장되어야 하며, 변경된 내용으로 웹서버 시스템에 반영시키려면 Control Panel에서 [Stop] 버튼을 클릭후 [Start] 해야 합니다. 즉 변경된 내용으로 반드시 다시 시작해야 한다는 겁니다.

2. 웹서버 루트 폴더 및 초기 파일

웹브라우저에 사이트 주소만을 입력하여 접속을 요청하면 그 사이트의 웹서버는 홈페이지의 시작 내용 즉 메인 페이지를 전송합니다. 이렇게 전송되는 메인 페이지를 초기 파일이라고 부릅니다. 그럼 이 초기 파일은 무엇이며, 어디에 저장되어 있을까요?

XAMPP에서는 초기 파일 이름과 이 파일이 저장된 폴더를 미리 설정해 놓고 있습니다. 이 폴더를 루트 폴더라고 하며 C:\xampp\htdocs 입니다. 이 루트 폴더에 초기 파일 이름이 저장되어 있는데 index.php라는 이름으로 설정되어 있습니다. 이러한 루트 폴더의 이름이나 경로 그리고 초기 파일의 이름을 웹서버 관리자는 재설정할 수 있습니다.

2.1 웹사이트 초기 파일 이름 변경

httpd.conf 파일을 열어서 DirectoryIndex 단어로 검색을 하면 다음과 같이 라인 286번에서 찾아줍니다. DirectoryIndex 단어 뒤에 index.php index.pl index.cgi 등 많은 파일들이 나열되어 있는데, 이 파일들이 바로 초기 파일이며 루트 폴더인 C:\xampp\htdocs 폴더에 저장되어야 합니다.

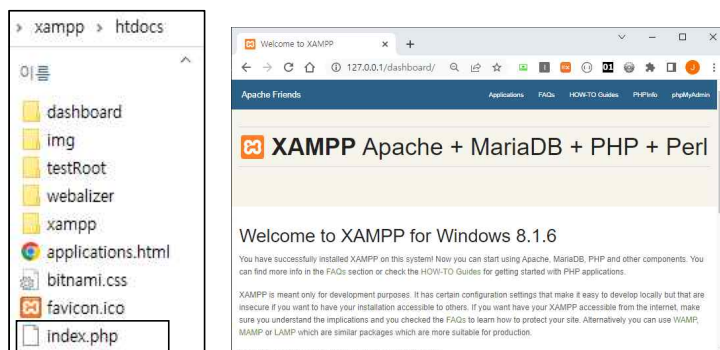
이 웹사이트에 브라우저로 접속하게 되면 웹서버는 C:\xampp\htdocs 루트 폴더로 이동하여 나열된 파일이 있는지 확인하고, 있으면 그 파일 또는 처리결과들이 웹클라이언트에게 전송됩니다.(Client Side면 파일을 전송하고, Server Side면 처리결과를 전송)

285	<IfModule dir_module>
286	DirectoryIndex index.php index.pl index.cgi index.asp index.shtml index.html index.htm W
287	default.php default.pl default.cgi default.asp default.shtml default.html default.htm W
288	home.php home.pl home.cgi home.asp home.shtml home.html home.htm
289	</IfModule>

모두 있을 필요는 없고 1개 파일만 있으면 됩니다. 2개 이상이 파일들이 C:\xampp\htdocs 루트 폴더에 있는 경우에는 가장 먼저 나열된 파일이 선택되어 처리결과가 전송됩니다. 즉 나열된 파일 이름들의 순서대로 선택 우선순위가 정해집니다. 만약 index.php 파일이 없다면 index.pl 파일이 선택되고, index.p 파일이 없다면 index.cgi 파일이 선택되고... 이렇게 선택됩니다. 나열된 파일 이름들은 웹 관리자만 알고 있는 내용입니다.

초기 파일이 필요한 이유는 사이트 주소만 입력해도 홈페이지의 내용을 확인할 수 있도록 하기 위해서입니다. 즉 사이트 주소 예를 들어 http://naver.com 주소만 입력해도 홈페이지의 내용이 보이는 이유는 등록된 초기 파일이 동작하기 때문입니다. 이렇게 하면 초기 파일의 이름을 외부에서 알 수 없도록 할 수 있습니다.

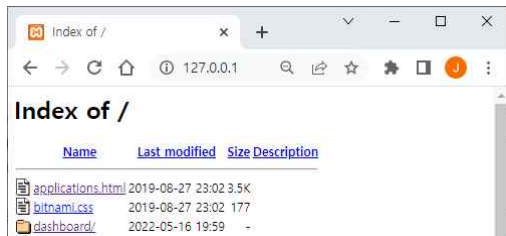
그럼 httpd.conf 파일을 변경해서 테스트 해보죠(변경된 httpd.conf 파일을 저장한후, 웹서버의 재시작은 필수입니다). 285 라인에 있는 index.php 파일이 있고, 이 파일이 초기 파일입니다. 웹브라우저에서 127.0.0.1 또는 localhost 접속하면 index.php 파일이 실행되었기 때문에 아래와 같은 내용이 출력된 겁니다. 입력하는데 좀 불편하지만 127.0.0.1/index.php 형태로 접속해도 문제 없겠죠?



(케이스 1) 여기에서 index.php 파일 이름을 삭제하고 접속하면 파일들의 목록들이 표시되는데 어쨌든 index.php 파일이 없어서 제대로 출력되지 않을 겁니다

286	DirectoryIndex index.pl index.cgi index.asp index.shtml index.html index.htm ₩
-----	---

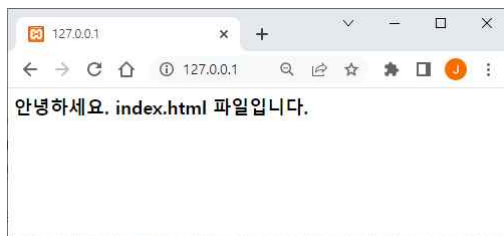
index.php 삭제하고 접속



(케이스 2) 이번에는 변경된 286 라인의 5번째 우선순위인 index.html 파일을 간단하게 만들어서 C:\xampp\htdocs 루트 폴더에 저장하고 127.0.0.1로 접속해 보겠습니다. index.html 파일이 5번째 우선순위가지만 앞 순위의 파일들이 없기 때문에 index.html 파일의 내용이 선택되어 출력됩니다.

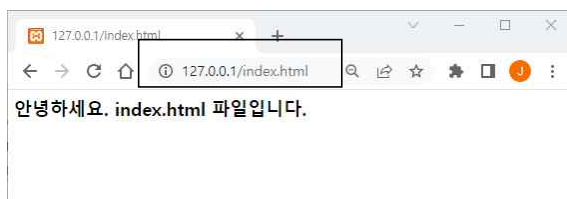
index.html 파일	<h2> 안녕하세요. index.html 파일입니다. </h2>
---------------	-------------------------------------

index.html 파일 생성후 접속



만약 목록에서 삭제했던 index.php 파일 이름을 원래대로 1순위로 등록하면 당연히 index.php 파일의 내용이 출력되겠지만 그럼에도 index.html 파일을 출력되게 하려면 어떻게 해야 할까요? 바로 주소창에 127.0.0.1 대신 127.0.0.1/index.html 방식으로 접속하면 됩니다.

286	DirectoryIndex index.php index.pl index.cgi index.asp index.shtml index.html index.htm ₩
-----	--



(케이스 3) httpd.conf 파일의 DirectoryIndex 부분에 나열된 파일 이름들만 사용해야하는 건 아닙니다. 웹사이트 개발자 또는 관리자가 임의대로 파일 이름을 만들 수 있습니다. 예를 들어 myhome.php, gwnu.pl, homepage.html, 등등. 외부에서 전혀 알 수 없겠죠?

2.2 웹서버 루트 폴더 재설정

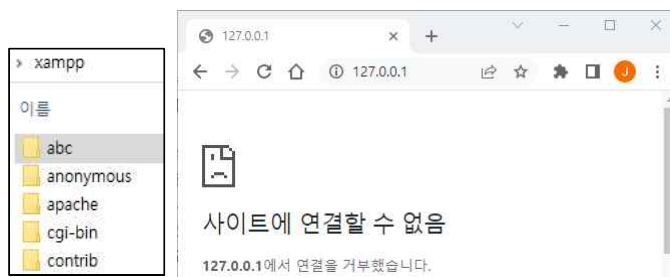
Apache 웹서버의 루트 폴더는 C:\xampp\htdocs 이며, 해당 사이트의 모든 홈페이지 관련

파일들은 이 폴더 및 그 하위 폴더에 저장됩니다. 그러나 이 루트 폴더는 공개된 정보이기 때문에 보안을 위해 변경하여 재설정할 필요가 있습니다.

httpd.conf 파일을 열어서 DocumentRoot 단어로 검색을 하면 다음과 같이 라인 252번에서 찾아집니다. 코드에서 보듯이 루트 폴더는 C:\xampp\htdocs 경로에 기본 값으로 설정되어 있습니다. 252번 라인만 변경해도 되지만, 253번 라인도 함께 수정해야 합니다.

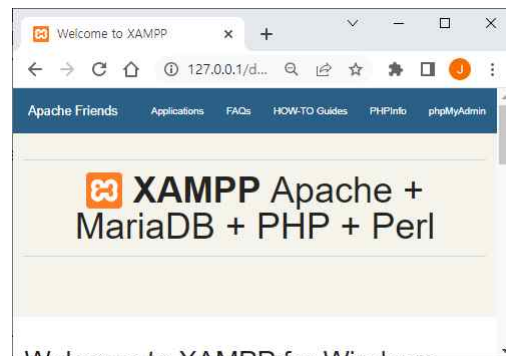
252	DocumentRoot "C:/xampp/htdocs"
253	<Directory "C:/xampp/htdocs">
:	:
:	:
279	</Directory>

(케이스 1) htdocs 루트 폴더 이름을 abc 이름으로 변경해서 웹 브라우저를 통해 127.0.0.1에 접속해 봅니다. C:/xampp/htdocs 폴더는 더 이상 존재하지 않기 때문에 접속되지 않습니다.



다음과 같이 DocumentRoot 부분의 C:/xampp/htdocs 폴더를 C:/xampp/abc 폴더로 변경해서 웹 브라우저를 통해 127.0.0.1에 접속해 봅니다. 이제 정상적으로 접속됩니다. 당연히 abc라는 폴더 이름으로 변경된 걸 외부에서는 전혀 모릅니다.

252	DocumentRoot "C:/xampp/abc"
253	<Directory "C:/xampp/abc">
:	:
:	:
279	</Directory>

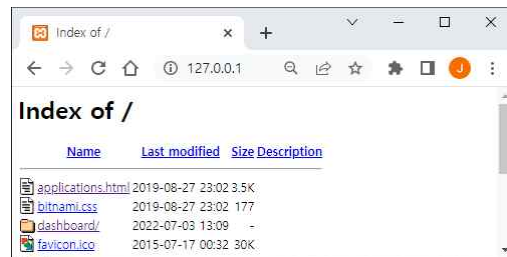


(케이스 2) 이번에는 C: 드라이브를 D: E: 와 같은 다른 드라이브에 홈페이지 파일을 저장해 놓겠습니다. 여기서는 H: 드라이브에 abc 폴더를 이동하고, httpd.conf 파일의 DocumentRoot 부분을 H:/abc으로 변경했습니다. 정상적으로 접속됩니다. 당연히 H: 드라이브 및 abc 폴더 이름으로 변경된 걸 외부에서는 전혀 모릅니다.

내 PC > 3_Education (H:)	252	DocumentRoot "H:/abc"
이름	253	<Directory "H:/abc">
abc	:	:
anaconda3	:	:
Cam	:	:
Data	279	</Directory>

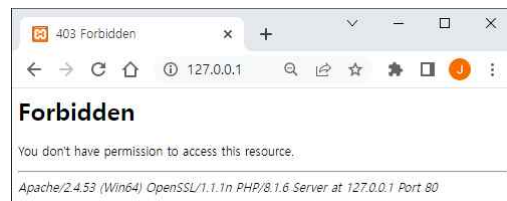
(케이스 3) 웹사이트 초기 파일 설정의 (케이스 1)에서 목록에 나열된 초기 파일이 없을 경우 C:\xampp\htdocs 루트 폴더에 저장되어 있는 모든 파일들의 목록이 표시됩니다. 이것을 디렉토리 인덱싱(Directory Indexing) 이라고 하는데, 이렇게 루트 폴더뿐만 아니라 하위 폴더들의 파일 목록을 제공하는 것은 보안에 매우 취약합니다. 이렇게 표시되는 이유는 278번 라인에 Require all granted 문장 때문입니다. 이 문장은 253번 라인에 있는 디렉토리(폴더) 목록표시를 허용한다는 의미입니다.

253	<Directory "C:/xampp/htdocs">
:	:
:	:
278	Require all granted
279	</Directory>



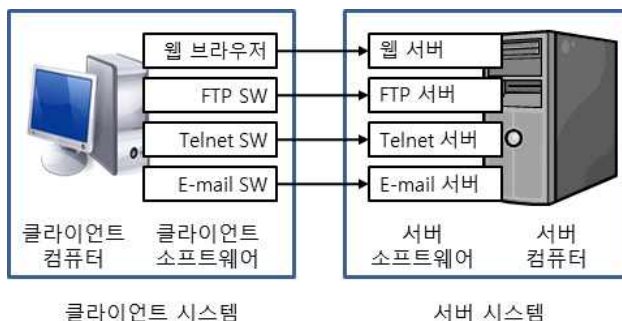
이 문제를 해결하기 위해 278번 라인을 허용한다는 의미의 granted 단어를 거부함을 의미하는 denied으로 변경하거나, #을 이용해 주석 처리하여 웹브라우저 접속하면 Forbidden 메시지가 표시되면서 파일 목록을 확인할 수 없게 됩니다.

253	<Directory "C:/xampp/htdocs">
:	:
:	:
278	Require all denied
278	# Require all granted
279	</Directory>



2.3 포트 설정

Chapter 01에서 설명했듯이 인터넷을 통해 활용할 수 있는 서비스들은 여러 가지가 있습니다. Web, FTP, Telnet, E-mail 등이 있죠. 이러한 서비스들은 하나의 서버시스템에 설치되어 동작될 수 있고, 또 실제로도 그렇게 운영하고 있습니다. 즉 하나의 컴퓨터에 여러 개의 서버 소프트웨어들을 설치하여 인터넷 서비스들을 제공할 수 있습니다.



그렇다면 인터넷 회선 하나로 여러 개의 서비스 즉 여러 개의 서버들에 접속할 수 있을까요? 이해를 돕기위해 케이블 TV를 생각해 보겠습니다. 케이블 TV는 하나의 회선으로 많은 채널을 통해 방송 콘텐츠를 제공합니다. 사용자는 원하는 방송 콘텐츠를 보기 위해 채널을 선택하게 됩니다.

이와 유사한 방식으로 인터넷 서비스들도 접속하게 됩니다. 인터넷 연결을 위해 컴퓨터에는 네트워크 카드가 있습니다. 이 네트워크 카드는 컴퓨터에 설치된 여러 서버에 접속될 수 있도록 포트(port)라는 것이 있고, 이 포트가 케이블 TV의 채널과 비슷한 역할을 합니다. 사용자가 웹서비스를 이용하려면 웹포트에 접속되고, FTP 서비스를 이용하려면 FTP 포트에 접속되는 거죠. 이렇게 접속되는 포트의 개수는 총 16비트 즉, 65,536개(0~65,535) 정도인데, 이중에서 Web, FTP, Telnet, E-mail 등의 인터넷 서비스를 위해 다음과 같이 포트 번호를 미리 할당해 놨습니다(시스템 포트).

인터넷 서비스	웹	FTP	Telnet	E-mail
포트 번호	80	21	23	25

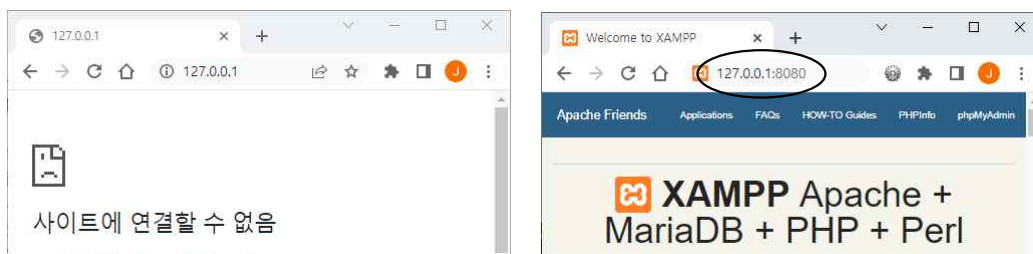
웹의 시스템 포트는 80번 이네요. 이 번호를 웹 주소 뒤에 콜론과 함께 :80을 추가하여 127.0.0.1:80 형태로 해도 접속이 됩니다. 원래 127.0.0.1로 접속하는 것과 같습니다.

이외에도 더 많은 시스템 포트 번호들이 있는데, 이 포트 번호들은 사용하면 안됩니다. 네트워크 포트와 시스템 포트에 대해서는 다른 자료를 참고하기 바랍니다.

그럼 웹 포트를 재설정해 보겠습니다. httpd.conf 파일을 열어서 Listen 단어로 검색을 하면 다음과 같이 라인 59번에서 찾아집니다. 59번 라인은 포트를 사용해서 접속하는 방법을 나타내는 주석문이고, 60번 라인에 80 이라는 웹 기본 포트를 원하는 포트 번호로 변경하면 됩니다. 여기서는 8080으로 재설정해 보겠습니다.

59	#Listen 12.34.56.78:80		59	#Listen 12.34.56.78:80
60	Listen 80	→	60	Listen 8080

수정된 httpd.conf 파일을 저장후 Apache 웹서버를 다시 시작해서, 127.0.0.1로 접속하면 다음과 같이 접속되지 않습니다. 웹 포트를 8080으로 재설정했기 때문에 127.0.0.1:8080으로 해야 접속이 됩니다.



다음과 같이 Listen 라인을 하나 더 추가하면 어떻게 될까요? 이렇게 하면 2개의 포트번호로 모두 접속이 가능합니다.

59	#Listen 12.34.56.78:80
60	Listen 80
61	Listen 8080

웹 포트 번호를 재설정해야 이유는 너무나 다양하기 때문에 일일이 설명하진 않겠습니다. 웹 서버를 관리하다 보면 이렇게 해야 하는 경우가 종종 발생하곤 합니다.

지금까지 httpd.conf 파일을 이용하여 Apache 웹서버 관리와 설정에 대해 알아 보았습니다. 웹서버 관리에 있어 가장 기본적인 내용이며, 보다 세부적인 설정에 대해서는 다른 자료들을 통해서 확인하기 바랍니다.

Chapter 04 PHP 동작의 이해

1. PHP 역사

PHP는 덴마크계 캐나다인 라스무스 러도프(Rasmus Lerdorf)가 1995년에 만든 것으로, 간편하게 웹사이트를 작성하기 위해 개발했습니다. 1995년 PHP 1.0 버전을 시작으로 개발이 계속되어 8.1.7 버전에 이르게 되었습니다.

처음에는 Personal HomePage tools의 줄임말이었는데, 현재는 Personal Hypertext Preprocessor로 사용하고 있습니다.

전세계 프로그래밍 언어 사용비율을 살펴보면 PHP는 13위 정도입니다. tiobe.com 사이트에서는 사용 비율에 대한 순위 정보를 매달 제공하고 있는데, PHP 언어는 10위권 내외에서 꾸준히 사용되고 있는 대표적인 웹사이트 개발언어입니다.

순위	프로그래밍 언어	사용비율
1	Python	12.20%
2	C	11.91%
3	Java	10.47%
4	C++	9.63%
5	C#	6.12%
6	Visual Basic	5.42%
7	JavaScript	2.09%
8	SQL	1.94%
9	Assembly language	1.85%
10	Swift	1.55%
11	Classic Visual Basic	1.33%
12	Delphi/Object Pascal	1.32%
13	PHP	1.25%

2022년 7월 기준/tiobe.com 발취

2. PHP 특징

PHP는 웹사이트를 제작하기 위해 만들어진 언어로써 HTML/CSS와 함께 사용됩니다. PHP 언어처럼 웹사이트를 제작하는 언어들은 ASP, JSP 등이 있습니다.

ASP(Active Server Page)는 마이크로소프트사의 프로그래밍 언어인 BASIC을 기반으로 만들어진 웹 프로그래밍 언어이며 마이크로소프트사의 IIS(Internet Information Service)라는 웹 서버에서만 동작합니다.

JSP는 Java 언어를 기반으로 해서 만들어진 웹 프로그래밍 언어이기 때문에 Java의 모든 기능을 사용할 수 있습니다.

보통 PHP는 중소 규모의 웹사이트 개발에 적합하고, JSP는 대규모 웹사이트 개발에 적합하다고 알려져 있습니다. 또한 PHP의 경우 연동되는 (Linux, Apache, MySQL)들이 무료이기 때문에 개발 및 비용은 ASP, JSP에 비해 저렴하다고 합니다.

다음의 표는 각 언어들의 최적화된 환경을 제시했지만 운영 상황에 따라서 다를 수도 있으니 참고만 하기 바랍니다.

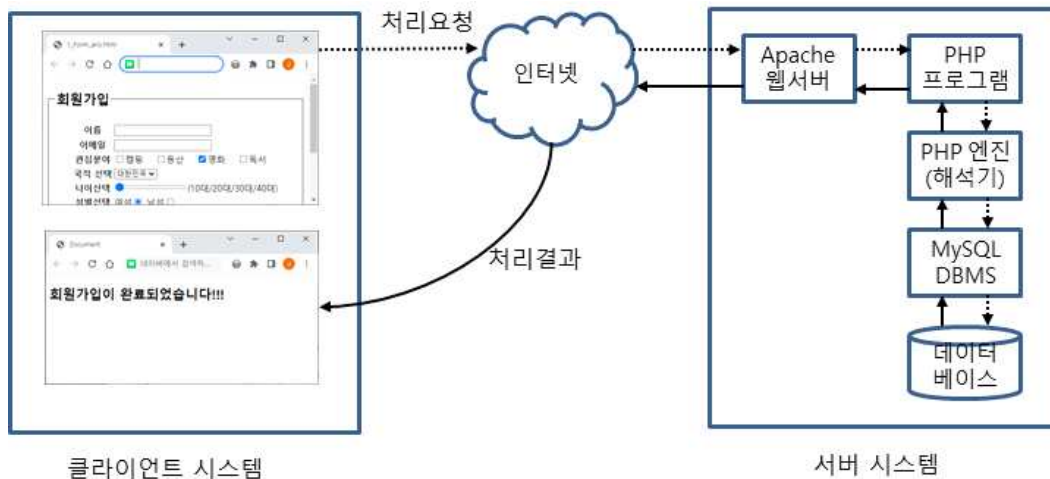
언어	운영체제	웹서버	DBMS
PHP	MS Windows, Linux, MacOS	Apache	MySQL
ASP	MS Windows	IIS	MSSQL
JSP	MS Windows, Linux, MacOS	Apache Tomcat	Oracle

PHP 언어의 특징에 대해 자세히 살펴보겠습니다.

- PHP 언어는 C 언어를 기반으로 해서 개발된 웹 프로그래밍 언어로써 C 언어의 기본 문법 및 제어 구조와 매우 유사합니다.
- PHP 엔진은 오픈소스이기 때문에 누구나 무료로 사용할 수 있습니다.
- PHP로 구축된 웹 사이트의 동작 속도가 빠르고 성능이 좋은 편이어서 접속자가 많아도 사이트 운영에 큰 문제가 없습니다.
- PHP 엔진은 다양한 하드웨어와 운영체제에서 사용되며 플랫폼 독립적입니다. Linux, MS Widows, MacOS 등의 운영체제제도 지원하므로 대부분의 플랫폼에서 사용할 수 있습니다.

3. PHP 웹 프로그램 처리 과정

PHP는 Apache와 같은 웹서버를 통해서만 실행이 가능하며, MySQL과 같은 DBMS와 연동해서 동작합니다. 따라서 PHP 프로그래밍을 위해서는 Apache, MySQL을 학습해야 합니다. 아울러 Front-end 기술인 HTML Form에 대해서도 기본적으로 학습되어 있어야 합니다.



위 그림의 회원가입 사례로 PHP 프로그램이 어떻게 동작하는지 설명하겠습니다.

- ① 서버 시스템에 회원가입을 신청하면 사용자의 웹브라우저에 회원가입 HTML Form이 표시됩니다.
- ② 사용자는 회원가입 폼에 내용을 입력하여 가입 관련 내용을 서버 시스템에 전송하여 처리 요청을 합니다.
- ③ 전송된 회원가입 정보는 Apache 웹서버를 거쳐서 미리 만들어 놓은 회원가입 PHP 프로그램에 전달됩니다.
- ④ 회원가입 PHP 프로그램은 PHP 엔진(해석기)에서 처리되어 MySQL DBMS를 통해 신규 회원으로 데이터베이스에 저장됩니다.
- ⑤ 여기서 회원가입 처리는 종료되지만, 사용자에게 잘 처리되었다는 메시지를 전달할 필

요가 있습니다. 이러한 안내 메시지는 클라이언트의 웹브라우저가 이해할 수 있어야 하기 때문에 HTML 형태로 전송되어야 합니다.

⑥ 이를 위해 PHP 프로그램은 HTML 태그를 전송할 수 있는 형태로 작성 됩니다. 이렇게 작성된 PHP 프로그램(회원가입 PHP 프로그램과 다를 수 있습니다)은 Apache 웹서버를 거쳐서 클라이언트의 웹브라우저에 전송됩니다. 또한 이 PHP 프로그램은 처리된 결과를 HTML 내용만으로 전송하기 때문에 소스코드가 공개되지 않습니다.

⑦ 클라이언트의 웹브라우저는 HTML 내용을 전달받아 사용자에게 보여줍니다.

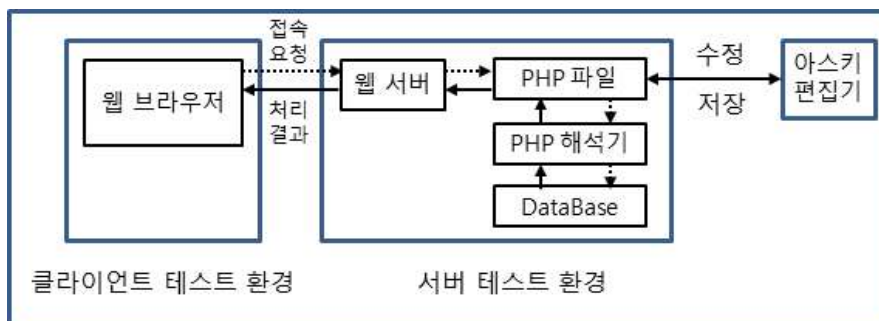
따라서 PHP 학습을 위해서는 기본적으로 HTML, 웹서버(Apache), 데이터베이스(MySQL)에 대해 알고 있어야 합니다.

4. PHP 프로그램 실행을 위한 테스트 환경

PHP 프로그램은 반드시 Apache 웹서버가 동작되는 시스템 환경에서 실행됩니다. 이를 좀 더 구체적으로 살펴보겠습니다.

우선 본 도서의 독자들은 대부분 한 대의 컴퓨터로 테스트를 진행할 겁니다. 따라서 한 대의 컴퓨터에 Apache 웹서버, PHP 엔진, MySQL 서버 등 서버 환경이 설치되어 있고, 동시에 웹브라우저와 같은 클라이언트 프로그램이 설치되어 있는 환경입니다.

컴퓨터 1대에서 PHP 프로그램 테스트



테스트 과정은 독자 여러분이 개발자 역할과 사용자 역할을 동시에 수행합니다. 사용자는 웹브라우저를 이용하여 PHP 프로그램에 접속하는 역할을 하게 됩니다.


4.1 PHP 프로그램 작성

PHP 프로그램은 어떤 ASCII 편집기로도 작성이 가능합니다. VS Code, Atom, Sublime 등 많은 ASCII 편집들이 있는데, 본 도서에서는 편집기에 대한 설명은 하지 않습니다. 대부분 무료이니 사용해 보고 자신에게 맞는 것을 선택하면 됩니다.

아직은 PHP 문법에 대해서 본격적으로 학습하는 단계가 아니기 때문에 단순히 PHP 소스 코드를 어떻게 테스트하는 것인지 이해만 하고 넘어가겠습니다.

① 우선 테스트를 위한 폴더를 생성합니다. 루트 폴더인 H:\xampp\htdocs 하위에 phptest 라는 폴더를 생성합니다.

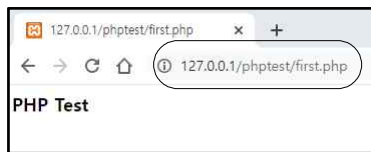
② 편집기를 사용하여 다음과 같이 간단한 PHP 코드를 작성해서 first.php 파일 이름으로 H:\xampp\htdocs\phptest 폴더에 저장합니다. 파일의 확장자는 반드시 php로 지정합니다.

first.php	<pre><?php echo "<h2>PHP Test</h2>"; ?></pre>	
-----------	---	--

4.2 PHP 프로그램 실행

반드시 웹서버가 동작중인 상태에서 웹 브라우저를 이용해 PHP 프로그램을 실행해야 합니다. 윈도우 탐색기에 있는 php 파일을 더블클릭하는 것은 PHP 프로그램의 실행이 아닙니다.

① 사용자는 웹 브라우저의 주소창에 127.0.0.1/phptest/first.php를 정확히 입력하여 PHP 프로그램에서 처리된 결과를 확인합니다.



혹시 『사이트를 연결할 수 없음』이라는 메시지가 나오면 Apache 웹서버가 동작하는지를 확인하세요. 그리고 『Not Found』라는 메시지가 표시되는 것은 폴더 이름이나 파일 이름을 잘못 표기한 경우입니다.

② 프로그램에 수정사항이 발생하면 ASCII 편집기를 이용해서 수정하고 저장후 웹 브라우저에서 확인합니다. 이런 과정을 반복하면 프로그램을 완성해 나갑니다.

이렇게 여러분들은 개발자 역할과 정보를 요청하는 사용자 역할을 동시에 수행하면서 PHP 프로그램의 테스트 과정을 진행합니다.

4.3 PHP 언어와 HTML과의 관계

Chapter 01에서 살펴본 것처럼 PHP 프로그램은 서버 시스템에 설치되어 있는 PHP 엔진에서 실행되는 Server Side Include입니다. 반면 HTML 파일은 서버 시스템에 저장되어 있지만 클라이언트 시스템에 다운로드되어 웹브라우저에서 실행되는 Client Side Include입니다. 이 내용들을 이해하고 있어야 PHP 언어와 HTML과의 관계를 파악할 수 있습니다.

4.3.1 PHP 프로그램 파일의 확장자

이미 알고 있듯이 PHP 파일의 확장자는 .php이고 HTML 파일의 확장자는 .html입니다. 이 파일의 확장자는 매우 중요합니다. 파일의 확장자가 무엇이냐에 따라 이 파일을 실행하는 소프트웨어가 결정됩니다. 예를 들어 확장자가 .hwp인 파일은 한글 소프트웨어로 실행되고, 확장자가 .xlsx인 파일은 Excel 소프트웨어로 실행됩니다. 너무 당연하죠.

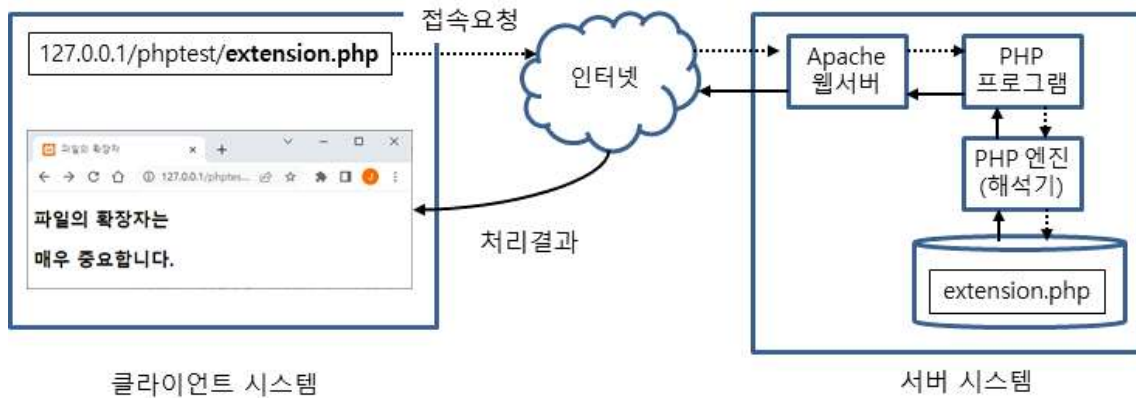
이렇듯이 확장자가 .php인 파일은 PHP 엔진에서 실행되고, 확장자가 .html인 파일은 웹브라우저에서 실행됩니다. 다음의 소스 코드를 보겠습니다. 동일한 소스코드의 내용을 extension.php, extension.html 확장만 달리해서 저장했습니다. 저장경로는 앞서 테스트한 경로인 C:/xampp/htdocs/phptest 폴더입니다.

extension.php	extension.html
<pre><!doctype html> <html> <head> <meta charset="UTF-8"> <title>파일의 확장자</title> </head> <body> <h2>파일의 확장자는 </h2> <?php echo "<h2>매우 중요합니다.</h2>"; ?> </body> </html></pre>	<pre><!doctype html> <html> <head> <meta charset="UTF-8"> <title>파일의 확장자</title> </head> <body> <h2>파일의 확장자는 </h2> <?php echo "<h2>매우 중요합니다.</h2>"; ?> </body> </html></pre>

이 파일들을 웹브라우저를 이용하여 접속 요청하면 다음과 같은 처리과정을 통해 그 결과가 웹브라우저에 표시됩니다.

먼저 extension.php 파일을 요청하기 위해 웹브라우저 주소창에 127.0.0.1/phptest/extension.php 입력합니다. 확장자가 .php인 파일은 PHP 엔진이 PHP 언어의 문법에 맞게 해석하고 웹브라우저에 처리 결과만 전달합니다. 예를 들어 파일에서 <?php ?> 부분을 PHP 엔진이 처리하여 결과인 <h2>...</h2> 코드만을 웹브라우저에 전달합니다.

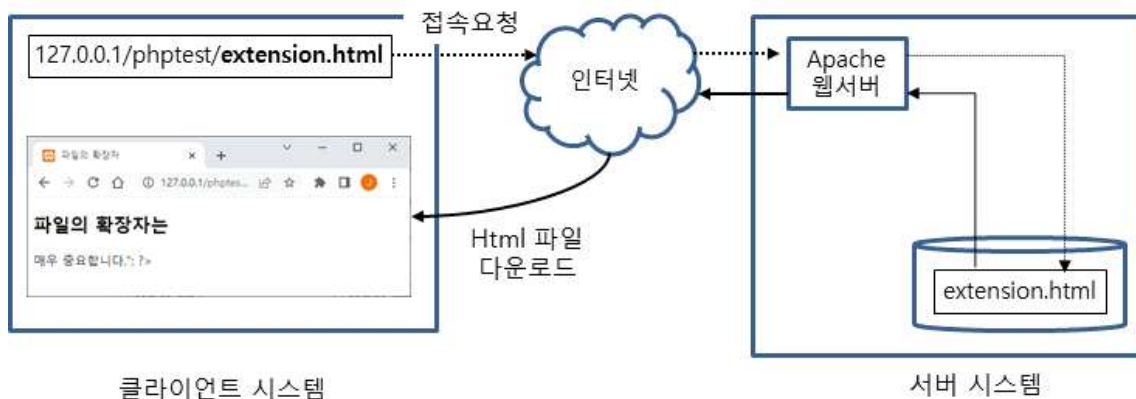
echo 명령문은 인용부호에 들어있는 내용을 웹브라우저에 그대로 전달하는 기능을 합니다.



위 웹브라우저의 결과화면에 표시된 내용을 『페이지 소스보기』로 확인하면 다음과 같이 표시됩니다. <h2>매우 중요합니다.</h2> 표시된 부분이 <?php ... ?> 부분을 PHP 엔진이 해석하여 웹브라우저에 전달한 코드입니다.

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>파일의 확장자</title>
</head>
<body>
  <h2>파일의 확장자는 </h2>
  <h2>매우 중요합니다.</h2>
</body>
</html>
```

이번에는 extension.html 파일을 요청하기 위해 웹브라우저 주소창에 127.0.0.1/phptest/extension.html 입력합니다. 확장자가 .html 파일은 클라이언트 시스템의 웹브라우저에서 실행되는 파일이므로 PHP 엔진의 처리없이 클라이언트 시스템에 다운로드됩니다. 다운로드된 파일은 웹브라우저에서 실행되는데 뭔가 이상하게 출력됩니다. <?php ... ?> 부분을 웹브라우저가 제대로 해석하지 못하기 때문입니다.



위 웹브라우저의 결과화면에 표시된 내용을 『페이지 소스보기』로 확인하면 다음과 같이 표시됩니다. extension.html 파일과 동일한 소스가 표시됩니다. <?php ... ?> 부분의 소스코드

가 그대로 노출됩니다. 서버쪽 소스코드를 노출시키는 건 보안상 매우 위험합니다. 따라서 비록 html 내용으로 작성된 파일이더라도 php 코드가 포함된 파일들은 반드시 .php 확장자로 관리해야 합니다.

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>파일의 확장자</title>
  </head>
  <body>
    <h2>파일의 확장자는 </h2>
    <?php
      echo "<h2>매우 중요합니다.</h2>";
    ?>
  </body>
</html>
```

이렇게 웹사이트 파일들의 확장자는 매우 중요하므로 반드시 주의가 필요합니다. 결론적으로 서버쪽에서 실행되는 파일의 확장자는 .php로 지정하고, 클라이언트쪽에서 실행되는 파일들은 .html로 지정해야 합니다.

4.3.2 HTML 구조에서 PHP 코드

이전에서 PHP 예제 파일은 HTML 파일의 기본 구조(<html>...</html>)를 갖추고 있습니다. 전체적으로는 HTML 코드이지만 그 속에서 PHP 코드가 삽입되는 형태인거죠. PHP 엔진에서 처리되는데는 아무 문제가 없습니다. 어떤 상황에서 이런 방식을 사용해야 한다는 원칙은 없습니다만 아마도 HTML 코드가 많은 상황에서 쓰인다고 볼 수 있습니다.

다음 코드의 파일 이름은 php_in_html.php이며 2개 숫자를 곱하기 하는 내용입니다. 이 파일을 웹 브라우저를 이용하여 요청하면 PHP 엔진에서 처리하여 곱하기 결과를 웹 브라우저에 표시합니다. 『페이지 소스보기』를 하면 곱하기 하는 수식은 나타나지 않고 곱하기 결과만 출력됩니다.

<pre><html lang="en"> <head> </head> <body> <?php \$op1 = 123; \$op2 = 456; \$multi = \$op1 * \$op2; echo "\$op1, \$op2 곱하기 결과 : \$multi"; ?> </body> </html></pre>	<div> <div> <div>웹브라우저 출력</div> <div>➡ 123, 456 곱하기 결과 : 56088</div> </div> <div> <div>페이지 소스보기</div> <div> <pre><html lang="en"> <head> </head> <body> 123, 456 곱하기 결과 : 56088 </body> </html></pre> </div> </div> </div>
--	--

<?php ... ?> 부분은 <body>와 </body> 사이에 작성되어 있습니다. 다른 위치에 있어도 실행에는 문제없지만 소스코드의 구조상 <body>와 </body>에 작성하는 것이 좋습니다.

4.3.3 PHP 파일 안에서 HTML 코드

다음 코드의 파일 이름은 `html_in_php.php`이며 첫 번째 라인부터 `<?php` 로 시작에서 `?>`로 마무리 됩니다. 그 안에서 2개 숫자를 곱하기 하는 코드들이 작성되었습니다. 이 파일을 웹 브라우저를 이용하여 요청하면 PHP 엔진에서 PHP 코드를 처리하여 곱하기 결과를 웹 브라우저에 표시합니다. 『페이지 소스보기』를 하면 곱하기 하는 수식은 나타나지 않고 곱하기 결과만 출력됩니다.

HTML 태그인 `<i>`가 `echo` 문을 통해 웹 브라우저에 전달됩니다. PHP 프로그램은 보통 이런 구조로 작성되는게 일반적입니다.

결론적으로 PHP 프로그램은 전체적으로는 PHP 구조로 작성하면서 웹 브라우저에 출력이 필요한 위치에서 `echo` 문을 이용해 HTML 태그나 출력 내용을 작성하면 됩니다.

<pre><?php \$op1 = 123; \$op2 = 456; \$multi = \$op1 * \$op2; echo "<i>"; echo "\$op1, \$op2 곱하기 결과 : \$multi"; ?></pre>	웹브라우저 출력 ➡ 123, 456 곱하기 결과 : 56088
	페이지 소스보기 <div><i>123, 456 곱하기 결과 : 56088</div>