PHP 10권 – 폴더 구조

**07. 폴더구조**

====================

이번장에서는 컴포저의 폴더 구조와 실제적으로 어떻게 연결이 되는지에 대해서 설명을 하도록 하겠습니다.

**07.1 기본 폴더**

컴포저를 설치하게 되면 vendor 폴더가 생성이 됩니다. 이곳에는 컴포저와 패키지 파일들이 설치되어 관리하게 됩니다.

Vondor 안에는 기본적으로 composer 폴더와 autoload.php를 가지고 있습니다. Vendor 폴더 안에 있는 autoload.php를 소스코드에 require 하여 사용하시면 됩니다.

<?php

// autoload.php @generated by Composer

require\_once \_\_DIR\_\_ . '/composer/autoload\_real.php';

return ComposerAutoloaderInitb0acc4ae67cab5585a791087586b8dcb::getLoader();

같이 생성이 되는 composer 폴더에는 자동으로 생성되는 오토로드 서브 파일들을 담고 있습니다. Vendor/autoload.php 파일을 보면 vendor/composer/autoload\_real.php 을 다시 참조합니다

**07.2 벤더명 폴더**

컴포저는 네임스페이스와 연동이 되어 패키지를 구별합니다. 네임스페이스라는 개념이 물리적인 디렉토리의 경로까지의 의미는 아닙니다. 하지만, 네임스페이스의 경로와 물리적 파일을 구조을 일치하여 사용하면 개발 유지하기 편리합니다.

컴포저는 vendor 디렉토리 안에 네임스페이스의 첫번째 이름으로 사용하는 벤더명 기준으로 서브 폴더를 생성합니다.

벤더명은 개발자, 회사의 고유 이름으로 서로 중복이 되지 않습니다.

**07.3 패키지 폴더**

네임스페이스 두번째 구분자는 패키지의 이름으로 사용을 합니다. 패키지의 이름은 자신의 벤더명 서브로 배포자가 적절하게 이름이 중복되지 않도록 지정을 하여 사용을 하면 됩니다.

패키지의 설치는 패키지 폴더 전체가 다운로드 되어 설치가 됩니다.

**07.4 서브 폴더**

패키지 안에는 실제 동작 코드 이외에 문서, 테스트 코드등 같이 포함하여 배포를 할 수 있습니다. 이런경우 패키지 안에서도 서브 폴더를 나누어 작업하여 배포를 하면 편리합니다.

주로 소스코드들은 src 폴더를 별도 추가하여 넣어 놓습니다. 패키지에서 서브 폴더를 구분하여 작업을 할 경우에는 composer.json 파일에서도 소스의 서브 경로를 같이 지정해 주어야 합니다.