**1 PHP 소개**

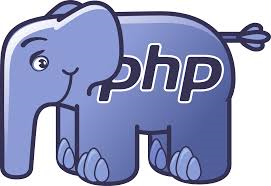
리눅스가 웹 서비스를 위한 운영체제로 인기를 끌기 시작되면서 PHP는 현재 7.x 버전까지 꾸준히 발전을 하면서 많은 개발자들의 사랑을 받은 오래된 웹 개발 언어입니다. 꾸준히 개발 현장에서 사랑을 받아온 PHP는 한때 인터넷 붐과 함께 잠시 주춤했으나 최근 페이스북 등 많은 소셜 사이트가 PHP로 개발한다는 소문에 다시 한번 인기를 얻고 있습니다. 또한 워드프레스, 두루팔 등 인기 있는 플랫폼들이 PHP로 개발되고 나아가 코드이그나이터, 라라벨 등 다양한 프레임 개발 환경까지 더해지면서 PHP 르네상스를 맞고 있습니다.



[사진출처 : 구글 이미지]

PHP는 웹 개발을 좀 더 편하고 빠르게 만들 수 있는 언어로, 빠른설치와 다른 언어보다 쉽게 배울 수 있습니다.

**1.1 PHP역사**



[사진출처 : 구글 이미지]

초기 PHP의 모태는 1995년 Rasmus Lerdorf에 의해서 PHP/FI라는 이름으로 시작했습니다. PHP/FI는 초기에 개인용 홈페이지 제작, 운영이 목적이었으나 이후 데이터베이스, 동적 웹 페이지 등 다양한 기능이 추가된 후 여러 사람들과 함께 소스를 공유하기 위해 공개되었습니다.

PHP/FI는 Perl과 비슷하게 변수, 폼 해석과 HTML 내장 문법을 갖고 있어서 현재 우리가 알고 있는 PHP의 기본적인 기능과 유사합니다. 다만 문법적으로 제한적이고, 단순하여 언어로서는 약간 일관성이 떨어지는 부족함이 있었습니다. 하지만 1997년 C 언어로 개발된 PHP/FI 2.0 버전은 많은 개발자들에게 인기를 얻으며 당시 웹 사이트의 약 1%의 50,000여 개의 웹 사이트가 PHP/FI 2.0으로 제작이 되었습니다.

1997년 Andi Gutmans와 Zeev Suraski에 의해서 PHP는 완전히 새롭게 재작성되어 PHP 3.0이 출시되었습니다. 새로운 PHP 3.0으로 탈바꿈되면서 기존의 PHP/FI 2.0은 개발이 중단되고, 요즘 들어 우리가 사용하고 있는 PHP의 모습을 띄게 됩니다.

PHP 3.0은 기존과는 다른 뛰어난 확장성이 있습니다. 일관성 있는 문법 체계와 객체지향 문법 체계를 도입함으로써 수많은 개발자 참여를 유도하고 새로운 모듈 개발이 활성화되기 시작했습니다. 이러한 환경들은 PHP를 웹 사이트를 개발하는 대표 언어로 발돋움하는 데 많은 기여를 하게 되었습니다. PHP 3.0은 출시한 지 9개월 만에 전 세계 웹 서버의 10%를 차지하는 성과를 얻으며 대중적인 웹 언어로 자리를 잡게 됩니다.

이러한 PHP 3.0의 성공으로 Andi Gutmans와 Zeev Suraski는 PHP 코드를 모듈화하고 복잡한 응용프로그램의 성능을 향상시키기 위해서 또 한 번의 대대적인 코어 재작성을 통해 2000년 5월 PHP 4.0을 출시하게 됩니다. 새로운 엔진 ‘Zend Engine’을 기반으로 향상된 성능뿐만 아니라 다른 핵심 기능들을 대거 추가하여 웹 서버에 대한 지원, HTTP 세션, 출력 버핑, 보안 등 새로운 언어 구조로 발전했습니다. PHP 3.0이 출시되고 2년 만에 나온 성과입니다.

PHP 5.0은 5년이라는 시간을 투자하여 2004년 7월에 발표되었습니다. 또 한 번 발전된 ‘Zend Engine2’ 엔진을 기반으로 PHP 5.0은 새로운 객체 모델과 새로운 기능들을 포함했습니다. PHP 5.0은 많은 PHP 4.0의 코드를 적은 리소스를 들여서 손쉽게 PHP 5.0으로 업그레이드하여 서비스를 유지할 수 있었습니다.

PHP 개발팀은 새로운 엔진과 성능 향상, 체계적인 문법적 언어 구조 등을 도입하여 또 다시 메이저 업그레이드 버전인 PHP 7.0 버전을 2015년 12월 발표되었습니다.

**1.2 서버 사이드**

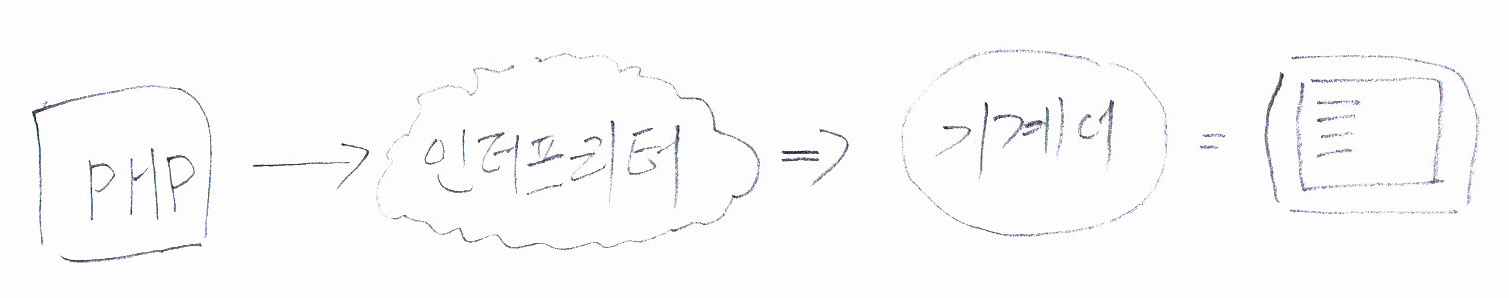
PHP는 대표적인 서버 사이드 언어입니다. 서버 사이드 언어란 프로그램 실행이 서버 쪽에서 동작을 하고 결과를 출력하는 형태의 언어를 말합니다. 서버 사이드 언어는 사용자의 작업을 서버에 요청 후, 서버가 요청된 처리를 한 후에 결과를 돌려주는 방식입니다.

PHP는 웹 서버와 같이 연동되어 사용자가 HTTP로 데이터 및 화면을 요청을 하면 서버에서 DB 연동 등 각종 데이터를 처리하여 화면을 사용자에게 보여주는 역할을 합니다. 대부분의 작업들이 서버 쪽에서 일어나기 때문에 서버 사이드 언어라고 합니다.

요즘에는 보다 빠른 서버 처리 및 대용량 처리를 위하여 화면 구성 등은 사용자 측에서 처리를 하고, 서버는 순수한 데이터를 제공하는 방식으로 변화를 하고 있지만 PHP가 서버상에서 돌아가는 언어라는 점에서는 서버 사이드 언어인 점은 틀림없습니다.

**1.3 인터프리터**

프로그래밍 언어는 대부분 프로그램 소스를 기계어로 번역, 생성하여 컴퓨터에 저장합니다. 이러한 기계어 번역 작업을 컴파일이라고 하는데, 대부분의 PC용 프로그램 등이 이러한 유형의 소프트웨어입니다. 대표적인 컴파일 언어로는 C 언어가 있습니다. 컴파일 언어는 매번 소스를 수정할 때마다 기계어로 번역해야 합니다.



인터프리터 언어는 이러한 컴파일 작업 없이 사용자가 실행을 요청할 때마다 코드를 해석 하여 처리 결과를 보여주게 됩니다. 소스를 자주 수정하거나 실행에 큰 부담이 없는 기능들은 대부분 인터프리터 형식의 언어로 많이 개발됩니다.

PHP는 컴파일 등 전처리 작업을 하지 않기 때문에 개발 속도가 매우 빠릅니다. 에디터에서 직접 PHP 코드를 삽입하면 즉시 해석하여 처리하게 됩니다.

**1.4 PHP 엔진**

PHP 엔진이란 PHP 스크립트을 분석하고 실행을 동작 처리하는 인터프리터 프로그램을 말합니다.

PHP 엔진은 C 언어로 개발되어 있습니다. 대표적인 PHP 엔진으로는 Zend 엔진과 페이스북에서 주도하는 HHVM이 있습니다.

PHP 엔진은 PHP 언어의 성능과 동작에 영향을 주는 실질적인 PHP의 핵심입니다. 최근에 PHP 엔진은 새롭게 바뀌고 성능 업데이트가 되면서 다시 한 단계 더 발전할 수 있는 기반을 마련하고 있습니다.

**1.5 젠드 오피캐시**

PHP 언어는 인터프리터 언어 입니다. PHP가 인기를 얻고 수많은 서비스에 적용되어 사용이 되면서 인터프리터의 태생적 한계인 소스처리에 대한 저리지연이 발생됩니다. 최근 들어 PHP는 캐시 적용 기술을 이용하여 PHP가 더 큰 용량의 서비스를 할 수 있는 다양한 기술들을 접목하고 있습니다.

C 언어 같이 전통적인 언어들을 소스 → 컴파일 → 실행 형태로 소스를 컴퓨터가 이해할 수 있는 기계어로 번역을 하고, 번역된 파일을 가지고 실행을 합니다. 윈도우 운영체제에서 확장자가 .exe로 되어 있는 파일처럼 기계어로 번역된 파일을 배포하고 사용자 컴퓨터에서 실행합니다.

하지만 PHP와 같은 인터프리터 언어는 스크립트를 실행할 때마다 소스를 분석하여 컴퓨터의 기계어로 번역을 처리하고 실제적인 실행 동작이 됩니다. 매번 소스 코드를 분석하고 실행 코드를 만들어내는 것은 컴퓨터의 리소스를 많이 소모하게 됩니다.

바이트코드 캐시란 PHP가 실행되면서 한 번 기계어로 번역한 것을 캐시로 저장하고 재사용하는 것입니다. 바이트코드 캐시를 이용하면 PHP는 서버의 부담을 줄여주고 보다 실행 속도를 개선하는 효과를 가져올 수 있습니다.

PHP는 이러한 바이트코드 캐시의 개념과 기능을 도입하여 젠드 오피캐시라는 이름으로 명칭했습니다.

젠드 오피캐시는 기본 설정 값이 아닙니다. 서버 담당자 또는 개발자가 별도의 설정으로 젠드 오피캐시를 활성화해야 합니다.

**1.6 PHP 미래**

PHP 언어는 발표 초기부터 한 시대를 주름잡았던 웹 개발 언어였습니다. 그러나 인기에 머물러 지속적인 업그레이드 지연 등 나태한 모습이었지만 지금은 다시 새로운 업데이트와 기능을 추가하여 제2의 전성기를 위해서 달려가고 있는 언어입니다. 최근 웹 서비스가 증가하면서 데이터 처리용으로 PHP 언어는 사용 빈도가 늘고 있습니다.

또한 모바일 시장의 폭발적인 성장과 더불어 웹 서비스의 백엔드 개발 작업으로 인기를 얻고 있습니다. 고적적인 웹 개발, 프레임워크 시스템, API 서비스 등 PHP는 지금 현재도 계속 발전하는 언어입니다.

또한 네임스페이스, 오토로딩 및 컴포저 등을 통해 최신 스타일의 언어와 개발 생태계를 만들어가고 있습니다. 웹 분야에서 PHP의 전성기가 다시 돌아올 것으로 예상합니다.