

이클립스 PDT
Eclipse PHP Development Tools 성명호 개발

이클립스 PDT

Eclipse PHP Development Tools 용영환 지음

지은이 **용영환**

펴낸이 **박찬규** | 엮은이 **김윤래** | 본문디자인 **복누리** | 표지디자인 **아로와 & 아로와나**

펴낸곳 **위키북스** | 전화 031-955-3658, 3659 | 팩스 031-955-3660

주소 경기도 파주시 교하읍 문발리 파주출판도시 535-7 세종출판벤처타운 #311

초판발행 2011년 03월 11일

ISBN 978-89-92939-72-0 13560

등록번호 제406-2006-000036호 | 등록일자 2006년 05월 19일

홈페이지 wikibook.co.kr | 전자우편 wikibook@wikibook.co.kr

이 책의 판권은 저자와의 독점 계약으로 위키북스가 소유합니다.
신 저작권법에 의해 한국 내에서 보호를 받는 저작물이므로 무단 전재와 복제를 금합니다.
이 책의 내용에 대한 추가 지원과 문의는 위키북스 출판사 홈페이지 wikibook.co.kr이나
이메일 wikibook@wikibook.co.kr을 이용해 주세요.

이 도서의 국립중앙도서관 출판시도서목록 CIP는
e-CIP 홈페이지 | <http://www.nl.go.kr/cip.php>에서 이용하실 수 있습니다.
CIP제어번호: CIP2011000682



이클립스 PDT

Eclipse PHP Development Tools 용영환 지음



“도움을 받았다면 최선을 다해서 갚아야 한다.”

존경하는 아버지께서 제게 하신 말씀입니다.

저는 지금까지 많은 문서와 오픈소스 프로젝트로부터 가치를 따질 수 없을 만큼의 도움을 받았습니다. 제가 알고 있는 지식이 원래 제 것이 아니기에, 제가 돌려드리는 방법 중 하나로 책을 쓰기로 했습니다.

이 책에는 제가 이클립스 PDT를 사용하면서 경험한 모든 것을 담았습니다. 최선을 다한 만큼 우리나라의 PHP 개발 환경에 조금이나마 도움이 되기를 진심으로 바랍니다.

함께 좋은 책을 만들자며 선뜻 손을 잡아주신 주신 위키북스 여러분과 박찬규님, 꼼꼼하게 읽고 좋은 의견을 보내주신 베타리더 여러분, 항상 잘 믿고 함께 해주시는 PHPKorea 여러분, 새로운 도전을 공감하고 실천할 수 있게 도와주신 큐브리드 여러분과 정병주 대표님, 그리고 이 순간에도 제게 지식을 나눠 주시는 이 세상 모든 분들께 고맙습니다.

이 책이 진짜 나오는지 확인하기 위해 결혼했다는 제 아내와 우리 가족 모두 사랑합니다.

2011년 2월 캐나다 토론토에서 용영환

<http://phpk.org>

E-Mail : ooti@phpkorea.org

김지한 : 소셜네트워킹 서비스 쿠(Kooo.net)를 개발했으며 현재 NHN UX팀에서 근무하고 있습니다.

어떤 언어를 이용해 소프트웨어를 개발할 때 좋은 산출물을 만들어내려면 당연히 개발자 본인이 문법과 구조, 언어의 특성을 잘 알고 있어야 합니다. 그리고 그런 것들을 가르쳐주는 책은 수없이 많습니다. 하지만 효율적인 개발 환경의 구성, 개발 및 디버깅 그리고 배포에 이르기까지 실무에서 실제로 요구되는 개발 기술은 아무도 가르쳐주지 않습니다. 흔히 能書不擇筆(능서불택필)이라고들 합니다. 그러나 개발자는 특히 橘化爲枳(귤화위지)이기도 합니다. 좋은 환경과 효율적인 방법은 훨씬 더 생산성 높은 개발을 가능케 합니다. 단지 PHP 문법이 궁금한 사람이라면 더 좋은 책이 얼마든지 있습니다. 하지만 제대로 된 PHP 소프트웨어 개발 방법을 가르쳐주는 것은 오직 이 책뿐이라고 자신 있게 권할 수 있습니다.

석종일 : NHN에서 오픈소스 DBMS인 CUBRID를 개발하고 있으며, 봄싹(springsprout.org)과 github.com/codeport를 통해 여러 가지 주제를 공부하고 있습니다.

서버에서 바로 수정하고 결과를 확인할 수 있다는 장점 때문에 간단한 웹 애플리케이션을 만들 때 PHP를 종종 사용했습니다. 그리고 서버에서 GUI 환경을 지원하지 않는 경우가 있어 개발 도구로는 그냥 vi를 이용했습니다. 하지만 이 책을 통해 이클립스가 자바뿐 아니라 훌륭한 PHP 개발 도구가 될 수 있다는 걸 알게 됐고 RSE를 이용해 손쉽게 원격 개발도 할 수 있게 되었습니다. 뿐만 아니라 이 책에는 유용한 플러그인이 많이 소개돼 있어 필요에 따라 하나씩 설치해서 사용하다 보면 쉽게 PHP 개발을 할 수 있을 것입니다.

오세형 : 네오플(neople.co.kr)에서 게임 유저들이 던전앤파이터를 제대로 즐길 수 있게 돕는 일을 하고 있습니다.

PHP 개발자에게 단비 같은 책이다. PHP 개발자라면 비주얼 스튜디오와 같은 걸출한 IDE를 한 번쯤은 꿈꿔봤을 테고, 이것저것 만져봤을 것이다. 이제는 방향할 필요가 없다.

김동현 : 휴대폰과 웹이 앞으로 어떻게 생활에 녹아들지가 궁금한 관련 업계 종사자입니다.

본업은 아니지만 취미로 PHP 코드를 작성할 때는 간단한 텍스트 에디터로 소스를 편집하고 그걸 서버에 올려서 다시 결과를 확인하는 과정을 반복했고, 비주얼 스튜디오처럼 개발 환경을 갖추지 않을까 생각을 많이 했습니다. 베타리딩 기간 동안 이클립스+PDT를 이용한 개발 환경 구축 및 PHP 개발을 좀 더 능률적으로 할 수 있는 방법에 대해 좋은 경험을 할 수 있었습니다. 아울러 저처럼 이클립스를 처음 접하는 분이라면 다양한 OS에서 쉽게 실습을 진행할 수 있게 쓰여 있어 많은 도움이 되었습니다.

김백기 : NHN에서 근무하며, 일본 네이버(naver.jp)를 개발하고 있습니다. 개인적으로 모바일 변환 사이트인 물조(<http://m.oolzo.com>)을 운영하고 있습니다.

2장의 디버깅은 꼭 보세요! 보통 디버깅은 가장 마지막에 부록으로 있는 경우가 많습니다. 사실 디버깅 실력이 프로그래밍 실력의 상당 부분을 차지하는데도 말이죠. 이클립스 PDT에서는 디버깅이 비교적 앞부분에 나와 있습니다. 실무에서는 디버깅이 가장 중요하기 때문에 이를 중요하게 여기신 것 같습니다. 더불어 실무 경험과 함께 깊은 통찰력도 느낄 수 있습니다.

한경훈 : 2009년 NHN 신입사원으로 입사해 뉴스 서비스 개발팀에서 근무했고 현재는 NHN Business Platform의 쇼핑 플랫폼개발팀에서 쇼핑 관련 업무를 개발하고 있습니다.

CHAPTER 01 들어가며

- 1-1. 우리에게 PHP란.....20
- 1-2. PHP 개발 환경.....21
- 1-3. 이클립스 PDT란.....22
- 1-4. PHP를 처음 배우는 학생들에게.....22
- 1-5. PHP 프로그램을 개발하고 있는 현업 개발자들에게.....23
- 1-6. 이 책의 대상 독자.....23
- 1-7. 시스템 요구사항.....24
- 1-8. 이 책의 구성.....24
- 1-9. 용어 정리.....25
- 1-10. 이클립스 아키텍처.....26

CHAPTER 02 이클립스 PDT 설치

- 2-1. JRE 설치.....28
 - JDK 다운로드.....29
 - 윈도우용 JDK 설치.....31
 - Mac OS X용 JDK 설치.....33
 - 우분투용 JDK 설치.....34
- 2-2. 이클립스 PDT 설치.....36
 - 윈도우에 이클립스 PDT 설치하기.....37
 - Mac OS X에 이클립스 PDT 설치하기.....39
 - 우분투에 이클립스 설치하기.....40

CHAPTER 03 이클립스 PDT 처음 시작하기

- 3-1. 이클립스 PDT 실행..... 44
- 3-2. 이클립스 PDT의 사용자 인터페이스..... 47
 - 퍼스펙티브..... 47
 - PHP 퍼스펙티브..... 48
 - 메뉴바와 툴바..... 49
 - 편집기..... 50
 - 뷰..... 50
- 3-3. PHP 프로젝트 생성..... 51
- 3-4. 기존 프로젝트를 이클립스 프로젝트로 가져오기..... 53
 - 기존 이클립스 프로젝트 가져오기..... 54
 - 일반 PHP 폴더를 이클립스 프로젝트로 만들기..... 55
 - 일반 프로젝트를 PHP 프로젝트로 만들기..... 56
 - Workspace 외부에 프로젝트 생성하기..... 57
- 3-5. PHP 파일 생성..... 58

CHAPTER 04 이클립스 PDT 살펴 보기

- 4-1. 콘텐츠 어시스트..... 62
 - 콘텐츠 어시스트 기능 활성화하기..... 63
 - 함수 어시스트..... 64
 - 클래스 어시스트..... 65

- 클래스 참조 어시스트..... 66
- 콘텐츠 어시스트 설정..... 68
- 콘텐츠 어시스트 아이콘..... 70
- 4-2. 풍선 도움말..... 70
 - 풍선 도움말 설정..... 71
- 4-3. 코드 자동 완성..... 72
- 4-4. 템플릿 설정..... 73
 - 템플릿 생성..... 73
 - 템플릿 변경 및 삭제..... 74
 - 템플릿 목록..... 75
- 4-5. 동일 코드 표시..... 77
 - 동일 코드 표시 설정..... 78
- 4-6. 새로운 확장자 등록하기..... 79
- 4-7. 작업 태그..... 81
 - 작업 우선순위 지정..... 83
 - 작업 태그 정의..... 84
- 4-8. PHP 도움말..... 85
 - 이클립스에서 PHP 온라인 도움말 바로 보기..... 85
 - 한글 PHP 온라인 도움말 보기..... 86
 - 오프라인용 PHP 도움말 보기..... 87

4-9. 북마크.....	89
북마크 위치로 바로 이동하기.....	90
4-10. 형식 계층.....	90
4-11. 오버라이드 표시.....	92
4-12. 실시간 에러 검출 기능.....	94
4-13. 철자 검사.....	95
4-14. 소스 코드 정리.....	96
4-15. PHP Include Path.....	97

CHAPTER 05 유용한 이클립스 기능 익히기

5-1. 외부 편집기 사용.....	104
5-2. 파일 비교.....	106
5-3. 편집기와 PHP Explorer 연결 보기.....	107
5-4. 퍼스펙티브 초기화.....	107
5-5. 단축키 확인 및 변경.....	108
5-6. PHP Explorer 파일 필터링.....	109
5-7. 워크스페이스 인코딩 및 개행 문자 변경.....	110
5-8. 외부 프로그램 실행.....	111
이클립스 변수.....	113

CHAPTER 06 실행과 디버그

6-1. Zend Debugger 플러그인 설치.....	117
업데이트 사이트를 이용한 설치.....	118
다운로드 설치 방법.....	120
Zend Debugger 설치 확인.....	121
6-2. PHP 스크립트 실행.....	123
PHP 실행 설정 변경하기.....	125
PHP 스크립트의 인자 값 전달.....	127
실행 결과를 파일로 저장하기.....	128
PHP 인터프리터의 버전 변경.....	129
Zend Debugger 플러그인에 모듈 추가하기.....	131
6-3. PHP 웹 페이지 실행.....	131
6-4. PHP 스크립트 디버깅.....	133
PHP Debug 퍼스펙티브.....	134
디버그 컨트롤.....	135
브레이크 포인트 지정.....	136
6-5. 로컬 웹 서버를 연동한 디버깅.....	137
웹 서버에 Zend Debugger 설치하기.....	137
Zend Debugger를 활용한 웹 서버 디버깅.....	139
리소스에 대한 경로 매핑.....	140

6-6. 원격 웹 서버를 연동한 실행.....	141
6-7. 원격 웹 서버를 연동한 디버깅.....	144
여러 개의 디버그 허용 IP 등록하기.....	147

CHAPTER 07 DMBS 연동

7-1. DBMS 연동 준비.....	151
7-2. GEF 설치.....	152
7-3. Quantum DB 설치.....	154
7-4. Quantum 퍼스펙티브.....	156
MySQL 연결.....	158
CUBRD 연결 방법.....	162
7-5. Quantum 사용법.....	166
테이블 생성 방법.....	167
테이블 삭제 및 전체 데이터 삭제.....	168
데이터 추가(INSERT).....	170
데이터 조건 및 정렬 변경.....	171
데이터 변경(UPDATE).....	172
데이터 삭제(DELETE).....	173
DB 인코딩 설정.....	174
데이터 변경이 안 되는 문제.....	175
결과 데이터를 파일로 저장하기.....	176

CHAPTER 08 형상 관리

8-1. SVN.....	180
Subclipse 설치.....	181
Subversive 설치.....	184
JavaHL과 SVNKit.....	188
SVN 저장소 연결.....	189
SVN 저장소에 프로젝트 생성하기.....	191
변경사항 커밋.....	195
변경사항 업데이트.....	197
프로젝트 체크아웃.....	198
소스 코드 되돌리기(Revert).....	201
8-2. 패치.....	204
패치 생성.....	204
Patch Root와 패치 적용 위치.....	206
패치 적용.....	206
서버에서 패치 적용하기.....	207
트렁크, 브랜치, 태그.....	209

CHAPTER 09 Ant를 이용한 손쉬운 배포

9-1. Ant 준비.....	212
Ant에 FTP 기능 설치하기.....	215
9-2. Ant로 FTP 전송.....	217
No grammar constraints detected for the document 경고 제거.....	220
FTP 전송 예외 처리.....	221
디렉터리 및 파일 삭제.....	222
변경된 파일만 FTP로 전송하기.....	223

CHAPTER 10 RSE를 이용한 원격 개발

10-1. RSE 설치.....	227
10-2. FTP를 이용한 원격 개발.....	229
10-3. RSE에서 PHP 파일 편집하기.....	232
10-4. 캐시 파일 동기화.....	232
RSE에서 PHP 콘텐츠 어시스트 사용하기.....	233
10-5. 원격 프로젝트 생성하기.....	234
10-6. SSH 연결.....	235

CHAPTER 11 Saros를 이용한 원격 협업 개발

11-1. Saros 설치.....	240
11-2. 구글 토크 서버를 통한 원격 개발.....	243
구글 토크 접속하기.....	244
친구 추가.....	245
전송 테스트.....	246
11-3. XMPP 서버를 통한 연결.....	246
11-4. 프로젝트 원격 공유.....	248
11-5. 실시간 협업.....	251

CHAPTER 12 Mylyn을 이용한 버그 트래킹

12-1. Mylyn 설치.....	254
12-2. Trac 연동.....	259
12-3. Mylyn 활용.....	262
작업(Task) 등록.....	262
작업 변경.....	264
작업 진행.....	266

CHAPTER 13 PHPDoc를 이용한 개발 문서 작성

13-1. PHPDoc 작성.....	270
13-2. phpDocumentor 설치.....	274
13-3. 이클립스에서 PHPDoc 생성하기.....	275
13-4. 템플릿.....	279
템플릿 인코딩 문제.....	279
템플릿 종류.....	279
13-5. 메모리 설정.....	281

CHAPTER 14 유용한 이클립스 플러그인

14-1. Notepad.....	284
14-2. Hex Editor.....	284
14-3. FreeMEM.....	285
14-4. RSS View.....	285
14-5. FileSync.....	286
14-6. StartExplorer.....	287

CHAPTER 15 이클립스 단축키

편집기.....	290
파일.....	291
내비게이터.....	291

부록 01 기타프로그램 설치

1-1. 웹 서버 설치.....	294
윈도우에 아파치 웹 서버 설치.....	294
1-2. 이클립스 PDT를 위한 아파치 웹 서버 설정.....	297
1-3. PHP 설치.....	298
1-4. XAMPP 패키지 설치.....	302
윈도우용 XAMPP.....	302
Mac OS X용 XAMPP.....	307
리눅스용 XAMPP.....	312
1-5. 큐브리드.....	314
1-6. 서버버전 및 Trac 설치.....	319
윈도우에 서버버전 서버 설치.....	319
윈도우에 Trac 설치.....	322
우분투에 서버버전 서버 설치.....	324
우분투에 Trac 설치.....	327

CHAPTER **n3**
PHP 개발자를 위한
이클립스 PDT

이클립스 PDT 처음 시작하기

이클립스 PDT 실행

이클립스 PDT의 사용자 인터페이스

PHP 프로젝트 생성

기존 프로젝트를 이클립스 프로젝트로 가져오기

PHP 파일 생성

이클립스 PDT는 일반적으로 자주 사용하는 텍스트 에디터나 우리에게 익숙한 통합 개발 환경과 생김새가 비슷하므로 쉽게 익숙해질 것이다. 하지만 도구를 잘 사용하려면 그 도구의 본질을 잘 이해하고 어떤 기능이 있는지 충분히 접해보는 것이 매우 중요하다. 이 장에서는 이클립스 PDT로 처음으로 PHP 프로그램을 개발하는 분들을 위해 이클립스 PDT를 실행하는 방법과 사용자 인터페이스, PHP 프로젝트 생성 방법 등을 순서대로 살펴보겠다.

3-1. 이클립스 PDT 실행

이클립스는 설치 마법사를 사용해서 설치하지 않으므로 이클립스를 실행하려면 매번 이클립스가 설치된 폴더를 열고 eclipse.exe 파일을 실행해야 한다. 이클립스를 실행할 때마다 폴더를 여는 것이 귀찮다면 “바로 가기” 아이콘을 바탕화면에 만들어 두는 것도 좋다.

이클립스를 사용하려면 당연히 JDK(또는 JRE)가 설치되어 있어야 하며 컴퓨터의 주 메모리가 최소 1GB 이상이어야 이클립스에서 원활하게 프로그래밍할 수 있다. 물론 주 메모리가 1GB 미만인 컴퓨터에서도 이클립스를 구동할 수는 있지만 경우에 따라 답답함을 느낄 수도 있다. 그리고 여유가 된다면 컴퓨터에 2GB 이상의 주 메모리를 장착하기를 적극 권장한다. 필자의 경우 주 메모리가 1GB인 경우와 2GB인 경우 그리고 그 이상인 경우에 느끼는 체감 속도의 차이가 적지 않았기 때문이다.

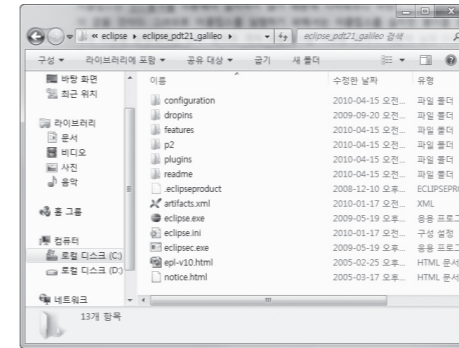
필자는 이클립스를 더 좋은 환경에서 구동하기 위해 모든 개발용 컴퓨터에 64비트 운영체제를 설치했다. 그 중에서도 64비트용 우분투의 만족도가 가장 뛰어났다. 우분투가 설치된 컴퓨터에 8GB 주 메모리를 사용했는데 아주 오랜 시간 켜두었음에도 개발 환경이 빠르고 안정적으로 유지됐다. 물론 윈도우 7 이상의 운영체제나 Mac OS X에서도 좋은 성능을 보이므로 자신에게 맞는 가장 좋은 환경을 찾아보기 바란다. 단, 이클립스 PDT는 64비트 윈도우용 버전을 제공하지 않으므로 꼭 32비트용 이클립스 PDT를 사용해야 한다.

자, 이제 이클립스 PDT의 세계로 들어가보자.

앞서 설치 과정에서 살펴본 바와 같이 이클립스 PDT는 압축 파일을 푸는 것으로 설치가 끝난다. 설치 마법사로 설치하지 않기 때문에 사용자는 이클립스 PDT 폴더에서 실행 파일을 직접 실행해야 한다.

이클립스 PDT를 실행하는 방법은 다음과 같다.

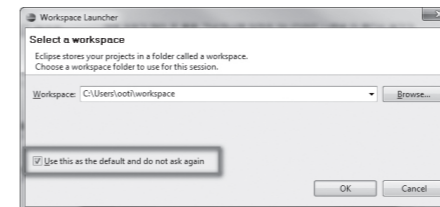
01. 윈도우 탐색기로 이클립스 PDT가 설치된 폴더를 연다.
02. 폴더에 있는 파일 중 eclipse.exe 파일을 찾아 실행한다.



[그림 3-1] 이클립스 PDT 설치 폴더

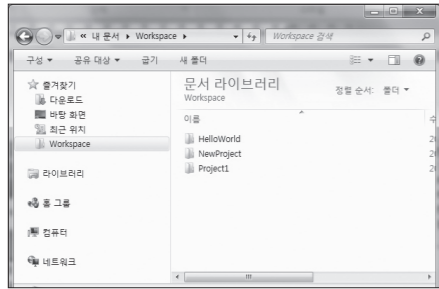
만약 .exe 확장명이 보이지 않는다면 윈도우 탐색기 메뉴의 도구 → 폴더 옵션 → 보기 화면에서 “알려진 파일 형식의 파일 확장명 숨기기” 체크를 해제하면 된다.

03. 이클립스 PDT를 처음 실행하면 기본 Workspace를 지정하기 위해 [그림 3-2]와 같이 Workspace Launcher 창이 열린다. 작업 폴더 경로를 직접 입력하거나 오른쪽의 Browse 버튼을 클릭해서 원하는 경로를 지정한 후 OK 버튼을 클릭한다.



[그림 3-2] Workspace Launcher 창

Workspace란 프로젝트 폴더를 비롯해 PHP 파일, HTML, CSS, JavaScript 파일 등을 작성하기 위한 최상위 작업 폴더를 말한다. 예를 들어 이클립스에서 HelloWorld라는 프로젝트를 새로 생성하면 [그림 3-3]과 같이 Workspace로 지정한 폴더 아래에 HelloWorld라는 폴더가 생성된다.

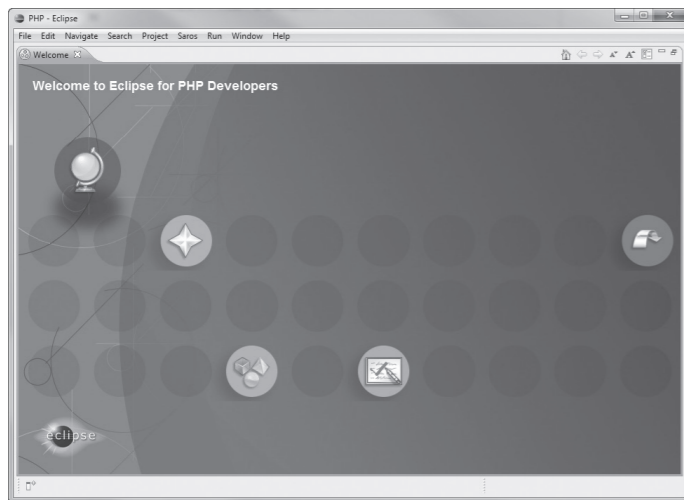


[그림 3-3] Workspace 폴더에 프로젝트 폴더가 생성된 화면

이클립스를 종료한 후 다시 실행하면 또 다시 Workspace Launcher 창이 열린다. Workspace로 사용할 폴더를 고정하고 다음에 실행할 때부터는 Workspace Launcher 창이 열리지 않게 하고 싶다면 “Use this as the default and do not ask again”에 체크를 한 후 OK 버튼을 클릭하면 된다.

Workspace 경로는 이클립스 상단 메뉴의 File → Switch Workspace를 차례로 선택해서 변경할 수 있다.

04. 이클립스 PDT가 정상적으로 실행되면 Welcome 화면을 볼 수 있다.



[그림 3-4] 이클립스 Welcome 화면

05. Welcome 탭의 X 버튼을 클릭하면 이클립스의 PHP 퍼스펙티브(Perspective)가 나타난다.

3-2. 이클립스 PDT의 사용자 인터페이스

이클립스 PDT의 사용자 인터페이스는 여러 개의 뷰(View)로 구성돼 있다. 각 뷰는 각기 다른 기능과 역할을 수행하는 플러그인이며, 필요에 따라 추가, 삭제, 이동 등 사용자가 원하는 대로 구성을 변경할 수 있다.

퍼스펙티브

이클립스의 퍼스펙티브(Perspective)란 뷰의 구성요소와 각 요소의 위치 등을 미리 정의해 둔 것을 말한다. 예를 들어 이클립스 PDT를 처음 실행했을 때 왼쪽에 PHP Explorer 뷰가 있고 오른쪽에 Outline 뷰가 있는 것도 PHP 퍼스펙티브에 각 뷰의 위치가 정의돼 있기 때문이다. 이클립스 PDT에는 PHP 퍼스펙티브와 PHP Debug 퍼스펙티브가 있으며 이클립스는 Web 퍼스펙티브, Java 퍼스펙티브 등 각종 작업을 위한 다양한 퍼스펙티브를 제공한다.

이 책에서 사용하는 PHP 퍼스펙티브는 PHP 프로그래밍을 하기 위한 편집기를 중심으로 구성돼 있으며 PHP Debug 퍼스펙티브는 디버깅을 위한 Debug 뷰, Breakpoint 뷰 등으로 구성돼 있다.

이 책에서 사용하는 퍼스펙티브는 다음과 같다.

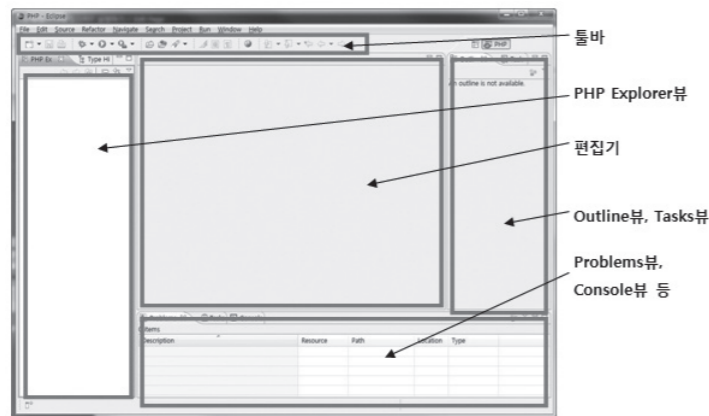
- PHP
- PHP Debug
- Remote System Explorer
- SVN Repository Exploring

현재 이클립스에 어떤 퍼스펙티브가 적용돼 있는지는 이클립스 상단 오른쪽에 퍼스펙티브 바에서 확인할 수 있으며, 다른 퍼스펙티브를 선택하면 선택한 퍼스펙티브에 정의된 형태대로 이클립스의 모습이 바뀐다.

PHP 퍼스펙티브

PHP 퍼스펙티브는 [그림 3-5]와 같이 다섯 개의 패널로 구성돼 있다. 이클립스 상단에 툴바가 있으며 가운데에는 편집기(Editor)가 있다. 왼쪽에는 PHP Explorer 뷰와 Type Hierarchy 뷰, 오른쪽에는 Outline 뷰와 Task 뷰, 그리고 이클립스 하단에는 Problems 뷰와 Tasks 뷰, Console 뷰가 자리잡고 있다.

이클립스 상단의 메뉴바와 툴바를 제외한 다른 뷰들은 패널 간에 자유롭게 이동할 수 있으며 숨기거나 보이게 할 수도 있다. 이동한 뷰의 위치는 뷰가 속해 있는 퍼스펙티브에 저장되므로 다음에 이클립스를 실행했을 때도 동일한 구성이 적용된다. 뷰를 옮기는 방법은 각 뷰의 제목을 마우스로 끌어서 원하는 패널에 놓으면 된다. 그리고 동일한 위치에 여러 개의 뷰가 있는 경우에는 자동으로 탭 형태로 보여지며 사용자는 원하는 탭을 선택해 뷰를 볼 수 있다.



[그림 3-5] 이클립스 PDT의 PHP 퍼스펙티브

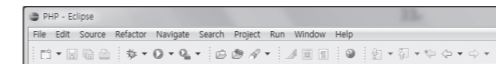


참고 퍼스펙티브 구성 초기화 방법

이클립스 상단 메뉴의 Window → Reset Perspective를 선택하면 퍼스펙티브를 초기화할 수 있다.

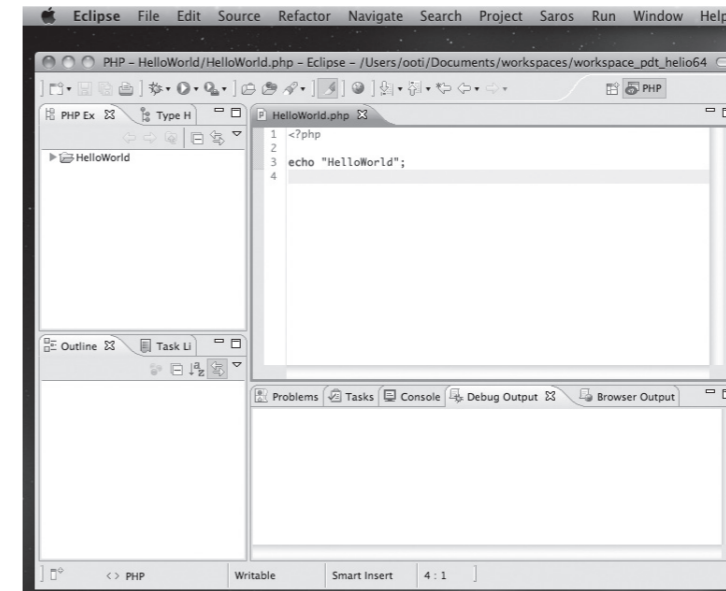
메뉴바와 툴바

이클립스 PDT 상단에는 메뉴바와 툴바가 위치해 있다. 메뉴바는 이클립스 PDT의 모든 기능을 분류별로 모아 놓은 인터페이스이며, 툴바는 이클립스 PDT에서 자주 사용하는 기능을 단축 버튼 형태로 모아 놓은 것이다. 툴바는 메뉴바와는 달리 퍼스펙티브에 따라 구성요소가 달라진다. 직접 메뉴바나 툴바의 구성요소를 바꾸고 싶다면 Customize Perspective 창에서 변경할 수 있다. Customize Perspective 창은 툴바 영역에서 마우스 오른쪽 버튼을 클릭한 후 Customize Perspective를 선택하면 된다.



[그림 3-6] 이클립스 PDT 메뉴바와 툴바

Mac OS X는 다른 운영체제와는 다르게 상단 메뉴바가 [그림 3-7]과 같이 분리돼 있다.



[그림 3-7] Mac OS X에서의 이클립스 메뉴바

편집기

이클립스 PDT의 가운데에 위치한 편집기는 소스 코드를 편집하거나 웹 프로젝트를 구성하는 각종 리소스(Resource)를 사용하기 위한 주 작업 영역이다. 이를테면, 여기서는 PHP나 XML 파일을 편집하거나 UML 다이어그램을 작성하기도 한다. 편집기는 다른 패널과 다르게 편집기 이외의 다른 뷰를 편집기 위에 올려둘 수 없다. 반대로 편집기도 다른 패널로 이동할 수 없다. 이것은 아마도 어떠한 경우에도 소스 코드를 편집할 수 있는 영역을 확보해 주려는 게 아닐까 싶다.

리소스는 프로젝트에 포함돼 있는 모든 파일을 말한다. 예를 들어 소스 파일을 비롯해 이미지 파일, 텍스트 파일 등이 리소스에 해당한다.

뷰

뷰(View)는 편집기를 중심으로 가장자리에 위치한다. 처음 이클립스 PDT를 실행하면 이클립스 화면 좌측에 PHP Explorer 뷰가 있고 우측에는 Outline 뷰 그리고 하단에는 Problems 뷰가 있다. 각 뷰는 리소스에 대한 정보나 프로젝트 또는 이클립스에 대한 여러 가지 정보를 시각적으로 보여준다. 이클립스 PDT를 실행했을 때 기본적으로 열려 있는 PHP Explorer 뷰는 PHP 프로젝트와 프로젝트 안에 포함된 폴더와 파일을 계층구조로 보여준다. 파일을 열거나 변경, 삭제할 때도 PHP Explorer 뷰를 이용한다. Outline 뷰는 편집기에 열린 파일에 대한 정보를 제공한다. PHP 파일인 경우에는 클래스, 함수, 변수 등의 목록을 한눈에 볼 수 있다. 예를 들어 목록에 표시된 함수를 클릭하면 그 함수가 정의돼 있는 위치로 편집기가 이동한다. Problems 뷰는 프로젝트에서 검출된 오류를 사용자가 알기 쉽게 보여준다. 이처럼 각 뷰는 제각기 담당하는 역할과 기능이 다르므로 뷰의 속성을 잘 이해하고 활용하는 것이 중요하다.

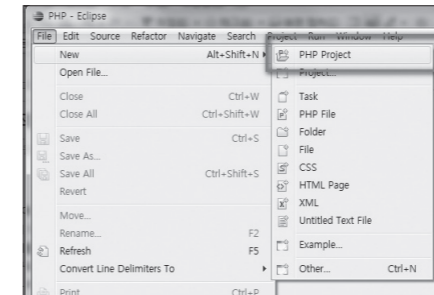
[표 3-1] PHP 퍼스펙티브 뷰

뷰	설명
PHP Explorer	일종의 파일 탐색기와 같은 역할을 수행한다. 프로젝트의 폴더나 파일을 관리할 수 있다.
Outline	편집기에 열린 PHP 파일의 클래스, 함수, 변수 등을 목록 형태로 보여준다.
Problems	소스 코드 및 프로젝트 환경 설정에 오류가 있는 경우 목록 형태로 정리해 보여준다.
Tasks	사용자가 해야 할 작업을 메모하거나 소스 코드에 포함된 작업 내용을 보여준다.
Console	PHP 프로그램을 실행하거나 기타 이클립스 작업 중 출력되는 메시지를 보여준다.
PHP Functions	PHP에서 제공하는 클래스와 함수를 목록 형태로 보여주므로 코드를 작성하면서 필요한 클래스나 함수를 찾는 데 유용하다.
Type Hierarchy	클래스의 계층 구조를 보여준다.
PHP Project Outline	선택한 프로젝트 전체에서 사용되고 있는 클래스, 함수, 변수 등을 보여준다. Outline 뷰가 편집기에 열린 파일에 대해서만 보여주는 반면 PHP Project Outline 뷰는 프로젝트의 전체 파일을 보여준다.

3-3. PHP 프로젝트 생성

이클립스 PDT에서 프로그램을 작성하려면 가장 먼저 프로젝트를 만들어야 한다. 프로젝트를 만들면 지정한 Workspace 폴더에 프로젝트와 이름이 같은 폴더가 만들어진다. 그럼 PHP 프로젝트를 만드는 방법을 살펴보자.

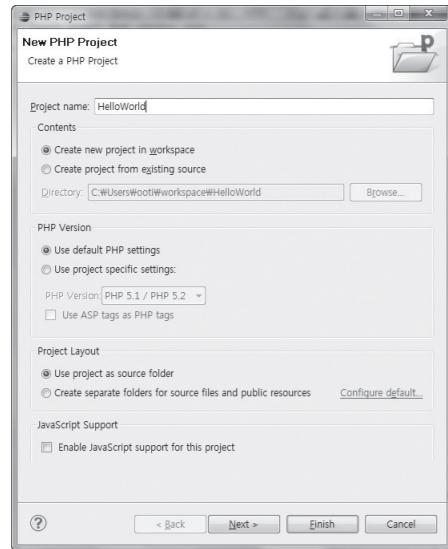
01. 이클립스 상단 메뉴에서 File → New → PHP Project를 선택한다. 만약 PHP Project가 보이지 않는다면 Other를 선택한 후 PHP Project 항목을 찾아 선택한다.



[그림 3-8] PHP Project 선택

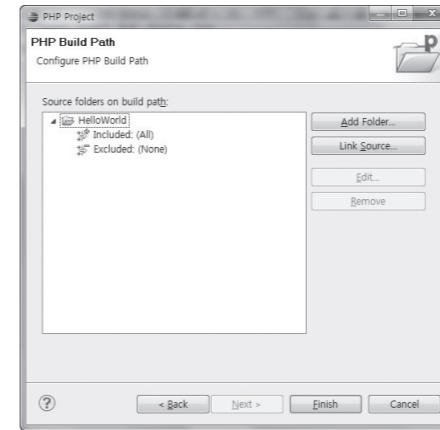
02. Project Name에 생성할 프로젝트의 이름을 입력한다. 이 장에서는 HelloWorld 프로젝트를 생성할

것이므로 Project Name에 HelloWorld를 입력한 후 Next 버튼을 클릭한다.



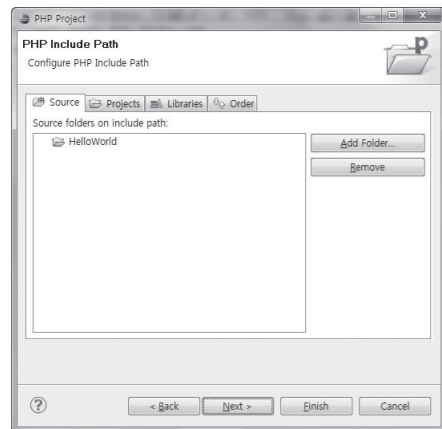
[그림 3-9] HelloWorld 프로젝트 생성

하면 프로젝트가 생성된다.



[그림 3-11] PHP Build Path 지정 화면

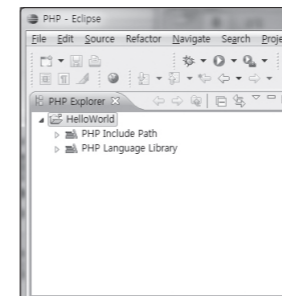
03. PHP Include path를 지정하는 화면이다. PHP Include path에 대해서는 나중에 자세히 알아볼 것이므로 일단은 Next 버튼을 클릭해서 다음 화면으로 진행한다.



[그림 3-10] PHP Include Path 지정

04. 프로젝트를 생성하기 위한 마지막 화면이다. PHP Build Path를 지정할 수 있다. Finish 버튼을 클릭

05. 프로젝트가 정상적으로 생성되면 이클립스 왼쪽 영역에 있는 PHP Explorer 뷰에 HelloWorld 프로젝트가 나타난다.



[그림 3-12] HelloWorld 프로젝트

3-4. 기존 프로젝트를 이클립스 프로젝트로 가져오기

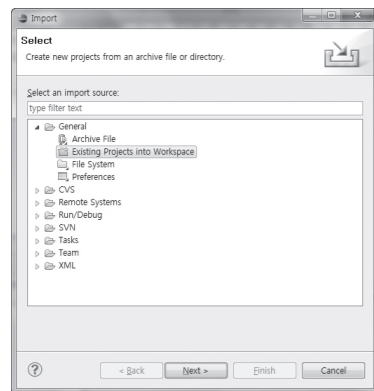
개발 도중에 이 책을 읽으면서 이클립스 PDT를 도입하려는 경우라면 기존에 개발하고 있던 PHP 프로그램을 이클립스에서 작업할 수 있게 가져오기(Import) 과정을 거쳐야 한다. 가져오기를 하지 않고 Workspace 폴더에 복사해 넣으면 이클립스가 인식하지 못하기 때문이다.

이클립스로 프로젝트를 가져오는 방법에는 여러 가지가 있다. 주의할 점은 가져올 프로젝트가 이클립스에서 작업하던 프로젝트인지에 따라 가져오는 방법에 다소 차이가 있다는 것이다. 가져올 프로젝트가 이클립스 프로젝트인지 확인하는 방법은 프로젝트 폴더를 열고 .project, .buildpath 등 이클립스 설정 파일이 있는지 확인하면 된다. 이클립스 프로젝트를 신규 프로젝트처럼 가져오면 기존에 저장돼 있던 설정값이 초기화될 수 있으므로 가져오기 작업을 하기 전에 확인해보길 바란다.

기존 이클립스 프로젝트 가져오기

기존에 이클립스에서 작업하던 프로젝트를 현재의 이클립스 프로젝트로 가져오려면 우선 기존 프로젝트 폴더 전체가 필요하다. 특히 .project 파일 같은 이클립스 프로젝트 설정 파일을 빼뜨려서는 안 된다.

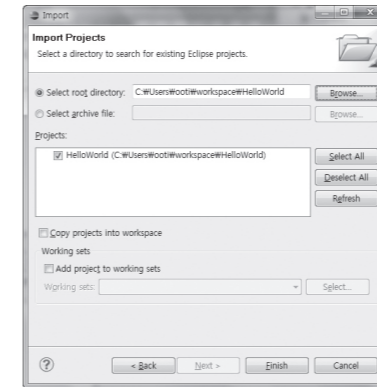
01. 이클립스 상단 메뉴에서 File → Import를 클릭한다.
02. Import 창에서 General → Existing Projects into Workspace를 선택하고 Next 버튼을 클릭한다.



[그림 3-13] 가져오기 대상 선택

03. Import Projects 화면에서는 가져올 이클립스 프로젝트 폴더를 지정한 후 Finish 버튼을 클릭한다. 기존 이클립스 폴더가 전혀 다른 폴더에 있어서 현재의 이클립스 Workspace로 복사해야 한다면 Copy projects into workspace를 체크하면 된다. 그리고 기존 프로젝트를 ZIP이나 TGZ 같은 압축 파일 형

태로 백업해둔 경우에는 Select archive file을 선택한 후 해당 프로젝트 백업 파일을 지정하면 자동으로 압축을 해제해서 Workspace로 가져온다.

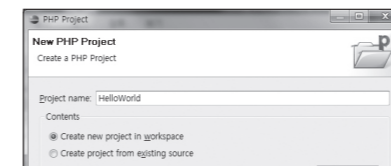


[그림 3-14] 가져올 프로젝트 지정

일반 PHP 폴더를 이클립스 프로젝트로 만들기

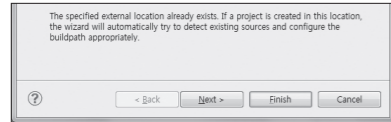
이클립스 PDT에서 작업하지 않던 일반 PHP 프로그램을 이클립스 PDT에서 개발하려면 이클립스 PDT의 PHP 프로젝트로 만들어 주는 과정을 거쳐야 한다. 이 장에서는 HelloWorld라는 일반 PHP 프로그램을 이클립스 PDT의 PHP 프로젝트로 변환하는 과정을 예로 들어 설명하겠다.

01. 일반 HelloWorld 프로그램의 폴더 전체를 이클립스의 Workspace 폴더 아래에 복사한다. 폴더 이름이 HelloWorld이라고 가정하자.
02. 이클립스 상단 메뉴에서 File → New → PHP Project를 차례로 선택한다.
03. Project name에 HelloWorld 프로젝트의 폴더 이름과 동일한 이름을 입력한다.



[그림 3-15] 가져올 프로젝트의 이름 입력

04. [그림 3-16]과 같이 폴더가 이미 존재한다는 메시지가 나타나면 무시하고 Finish 버튼을 클릭한다.



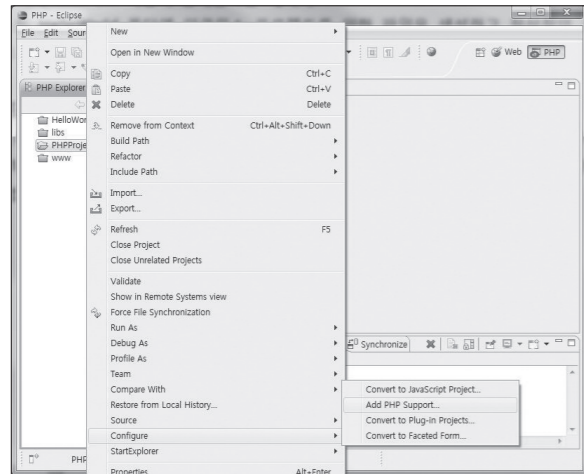
[그림 3-16] 같은 프로젝트 이름이 존재한다는 메시지

이클립스는 HelloWorld 폴더에 이클립스 프로젝트용 파일을 생성하고 정상적인 프로젝트로 인식한다.

일반 프로젝트를 PHP 프로젝트로 만들기

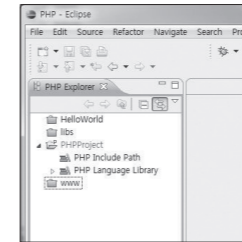
이클립스에서 일반 프로젝트나 HTML 프로젝트 등과 같이 다른 형식의 프로젝트에서 PDT의 기능을 사용하고자 한다면 해당 프로젝트를 PHP 프로젝트로 변경해야 한다. PHP 프로젝트로 변경하는 방법은 아주 간단하며 다음과 같은 방법으로 바꿀 수 있다.

01. PHP Explorer에서 PHP 프로젝트로 변경하고자 하는 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭한다.
02. 메뉴에서 Configure → Add PHP Support를 선택한다.



[그림 3-17] Add PHP Support 메뉴

03. PHP Explorer에서 변경한 프로젝트를 펼쳤을 때 PHP Include Path, PHP Language Library가 보이면 PHP 프로젝트로 변경된 것이다.

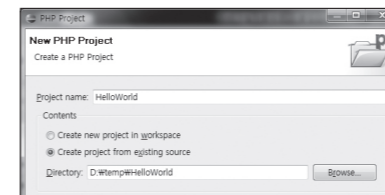


[그림 3-18] PHP 프로젝트로 변경된 프로젝트

Workspace 외부에 프로젝트 생성하기

기본적으로 이클립스에서 프로젝트를 생성하면 Workspace 폴더 아래에 생성되지만 Workspace가 아닌 다른 위치에 프로젝트 폴더를 생성해야 할 때도 있다. 이런 경우에는 다음과 같이 프로젝트를 생성하면 된다.

01. 이클립스 상단 메뉴에서 File → New → PHP Project를 차례로 선택한다.
02. New PHP Project 창이 나타나면 Project name을 입력하고 바로 Contents 항목에서 Create project from existing source를 선택한다.



[그림 3-19] Create project from existing source로 PHP 프로젝트 생성

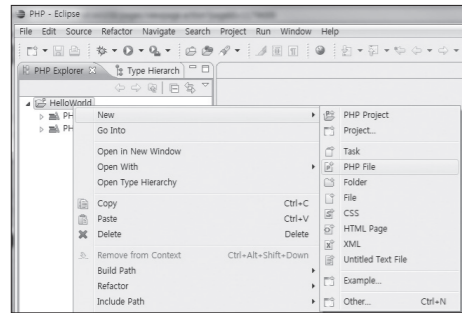
03. Directory 란이 활성화되면 빈 폴더를 만들고 해당 폴더를 지정한다.

3-5. PHP 파일 생성

PHP 프로젝트를 생성했다면 이제 PHP 파일을 만들어 보자. PHP 파일도 File 메뉴를 통해 생성할 수 있지만 여기서는 PHP Explorer 뷰에서 PHP 파일을 생성하는 방법을 살펴본다.

PHP 파일을 만드는 방법은 다음과 같다.

01. PHP Explorer 뷰에서 앞서 만든 HelloWorld 프로젝트를 선택하고 마우스 오른쪽 버튼을 클릭한다.
02. PHP Explorer의 메뉴가 나타나면 New → PHP File을 선택한다.

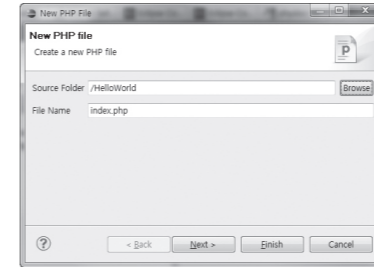


[그림 3-20] PHP Explorer 문맥 메뉴

03. New PHP File 창이 뜨면 File Name에 원하는 파일 이름을 입력한 후 Next 버튼을 클릭한다. 필자는 index.php라고 입력했다.

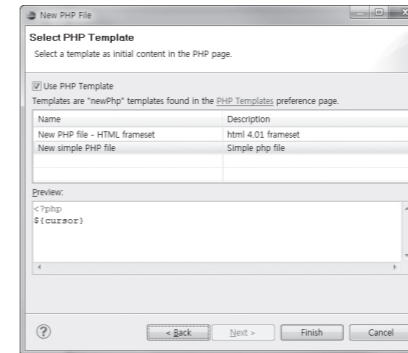
HelloWorld.php와 같은 다른 파일명을 입력해도 무방하나 일반적으로 웹 서버를 통해 가장 처음 실행되는 PHP 파일이 index.php라서 index.php라고 지정했다.

Source Folder에는 PHP 파일을 만들 폴더 경로를 지정하고 Next 버튼을 클릭한다. [그림 3-20]에서 HelloWorld라는 최상위 폴더를 선택했기 때문에 Source Folder에 /HelloWorld라고 입력돼 있음을 확인할 수 있다. Source Folder를 변경하고 싶다면 작업 경로를 입력하거나 Browse 버튼을 클릭해서 폴더를 선택하면 된다.



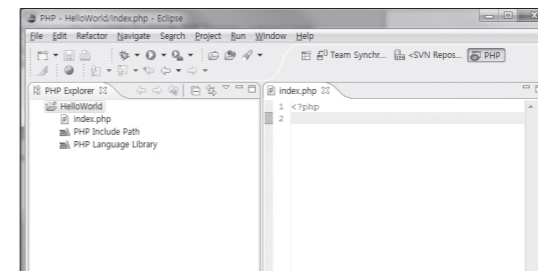
[그림 3-21] New PHP File 창의 PHP 파일명 입력 화면

04. PHP 템플릿을 선택하는 화면이다. 기본적으로 두 개의 템플릿이 준비돼 있다. New PHP File—HTML frameset 템플릿은 HTML 화면을 프레임으로 나누고 싶을 때 사용하는 템플릿이다. New simple PHP file 템플릿은 가장 간단한 템플릿으로서 PHP 코드를 작성하기 위한 최소한의 코드인 <?php만 담고 있는 템플릿이다. 여기서는 New simple PHP file을 선택한 후 Finish 버튼을 클릭한다.



[그림 3-22] PHP 템플릿 선택 화면

05. PHP 파일이 생성되며 PHP Explorer 뷰에서 확인할 수 있으며 자동으로 편집기에 파일이 열린다.



[그림 3-23] PHP 파일이 생성된 화면

06. 편집기에 열린 index.php 파일에 코드를 입력하고 저장한다. 파일을 저장하는 방법은 이클립스 상단 메뉴에서 File → Save를 선택하거나 **Ctrl+S**를 누르면 된다.

CHAPTER **n6**
PHP 개발자를 위한
이클립스 PDT

실행과 디버깅

Zend Debugger 플러그인 설치

PHP 스크립트 실행

PHP 웹 페이지 실행

PHP 스크립트 디버깅

로컬 웹 서버 연동 디버깅

원격 웹 서버를 연동한 실행

원격 웹 서버를 연동한 디버깅

프로그램을 개발하기 위해 우리는 최대한 실제 동작 환경과 비슷한 개발 환경을 준비한다. 웹 프로그램은 개발 과정에서 웹 서버에서 구동되는 프로그램의 상태를 지속적으로 확인해야 하기 때문에 일반적으로 로컬 컴퓨터에 웹 서버와 DBMS 서버를 설치해서 프로그램을 개발한다. 개발 시스템에 여유가 있으면 테스트용 웹 서버를 별도로 마련해서 개발을 진행하기도 한다. PHP 웹 프로그램을 개발할 때 웹 서버가 꼭 필요한 것으로 여겨지기도 하지만 사실 전체 웹 프로그램에서 반드시 웹 서버를 사용해야만 하는 부분은 많지 않다. 굳이 웹 서버와 웹 브라우저를 통해 실행하고 디버깅할 필요가 없는 부분을 개발할 때는 이클립스 PDT의 실행과 디버깅 기능을 활용하는 편이 경제적이다. UI에 영향을 주지 않는 코드를 수정했는데도 매번 웹 브라우저를 통해 테스트하는 것은 것은 전체적인 개발 속도를 떨어뜨리는 요인으로 작용할 수 있다.

이클립스 PDT는 웹 프로그래밍 언어라는 PHP의 특성이 잘 반영된 개발 도구다. 이클립스 PDT에서는 로컬 컴퓨터에 웹 서버를 설치하지 않은 상태에서 PHP 스크립트를 실행하고 테스트할 수도 있고 웹 서버와 연동해서 PHP 프로그램을 실행할 수도 있다.

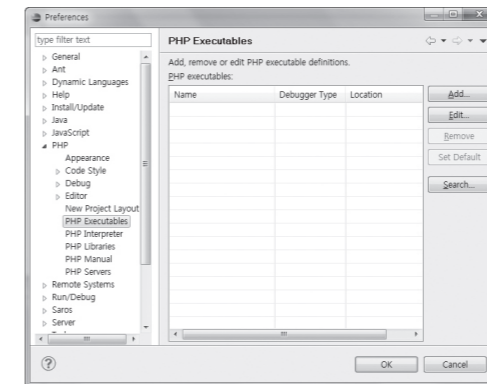
특히 디버깅 기능을 사용하면 그동안 웹 프로그래밍을 디버깅하면서 느꼈던 답답함이 사라질 것이다. 이클립스 PDT에서 디버깅할 경우 소스 파일 안에 디버깅용 코드를 작성하고 디버깅이 끝나면 해당 코드를 다시 지우는 작업을 되풀이하지 않아도 된다. 또한 실제 웹 서버에서 구동되는 PHP 프로그램까지도 디버깅할 수 있다. 더불어 웹 브라우저가 GET 방식으로 전달하는 값인 QUERY_STRING 상수 값까지도 이클립스 PDT에서 직접 정의할 수 있다.

이클립스 PDT는 PHP Script Run/Debug와 PHP Web Page Run/Debug으로 네 가지 실행 방법을 제공한다. 로컬 웹 서버 연동 실행, 로컬 웹 서버 연동 디버깅, 원격 웹 서버 연동 실행, 원격 웹 서버 연동 디버깅은 이클립스 PDT의 PHP Web Page 기능을 사용하지만 사용법에 다소 차이가 있으므로 경우에 따라 각 사용법을 이해하는 것이 중요하다.

자, 그럼 이클립스에서 PHP 프로그램을 어떻게 실행하고 디버깅하는지 자세히 살펴보자.

6-1. Zend Debugger 플러그인 설치

이클립스에 PHP 디버거가 설치돼 있지 않다면 PHP 프로그램을 실행하거나 디버깅할 수 없으므로 꼭 설치해야 한다. PHP 디버거가 이클립스에 설치돼 있는지 확인하는 방법은 이클립스 상단 메뉴에서 Window → Preferences를 선택한 후 [그림 6-1]과 같이 PHP → PHP Executables에서 확인할 수 있다.



[그림 6-1] PHP 디버거 설정 창

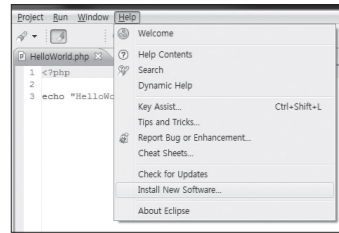
Zend Debugger는 Zend사에서 만든 PHP용 디버그 프로그램으로서 웹 서버에서 PHP 인터프리터와 함께 사용할 수 있는 PHP 모듈과 이클립스 플러그인으로 두 가지 형태로 배포되고 있다. 이어서 두 가지 버전을 모두 설치해서 사용하겠지만, 우선 이클립스 플러그인 형태의 디버거에 대해 살펴보자.

PHP Executables는 PHP 실행기 또는 PHP 인터프리터라는 용어로 부르는 것이 정확하다고 생각한다. 그럼에도 이 책에서 굳이 PHP 디버거라고 하는 이유는 이클립스 플러그인 형태의 PHP 인터프리터에 기본적으로 PHP 디버거가 포함돼 있고, 주 사용 목적이 디버깅이기 때문이다. 더불어 Zend사의 다운로드 페이지에도 PHP Debugger라고 표기돼 있다.

이클립스용 Zend Debugger 플러그인을 설치하는 방법에는 두 가지가 있다. 하나는 업데이트 사이트를 통해 자동으로 설치하는 방법이고, 다른 하나는 직접 다운로드 페이지로 들어가서 Zend Debugger 플러그인을 내려 받아 설치하는 방법이다. 두 가지 방법 모두 특별히 어려운 점은 없으므로 선호하는 방법으로 설치하면 된다.

업데이트 사이트를 이용한 설치

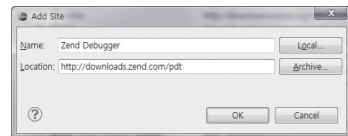
01. 이클립스 상단 메뉴에서 Help → Install New Software를 차례로 선택한다.



[그림 6-2] Help → Install New Software 선택

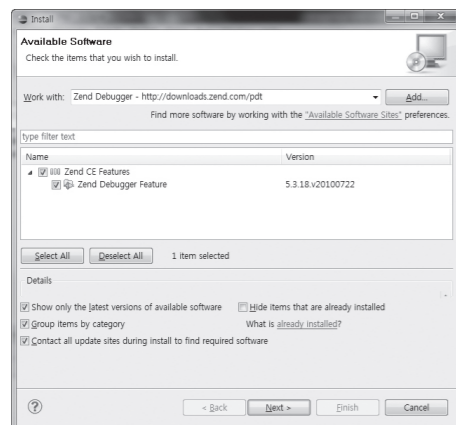
02. Install 창이 뜨면 Add 버튼을 클릭한다.

03. Add Site 창의 Name에는 Zend Debugger를 입력하고 Location에는 http://downloads.zend.com/pdt를 입력한 후 OK 버튼을 클릭한다.



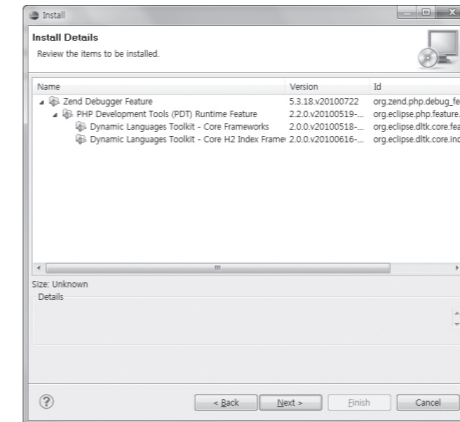
[그림 6-3] Add Site 창

04. Zend Debugger 항목을 선택한 후 Next 버튼을 클릭한다.



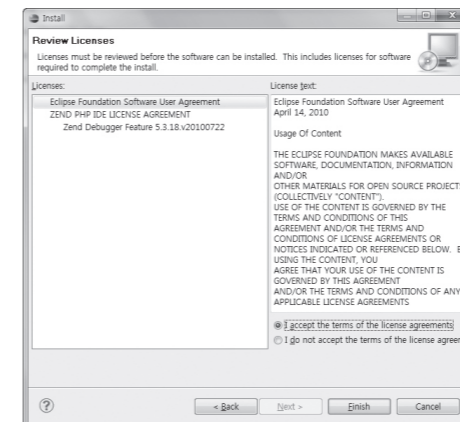
[그림 6-4] Install 창 설치 항목 선택

05. 설치할 항목을 자세히 보여주는 화면이 나타난다. 확인 후 Next 버튼을 클릭한다.



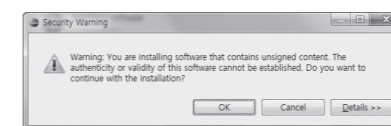
[그림 6-5] Zend Debugger 설치 항목 확인

06. 라이선스 동의 화면이 나타나면 I accept the terms of the license agreements 항목을 선택해서 라이선스에 동의한 후 Finish 버튼을 클릭한다.



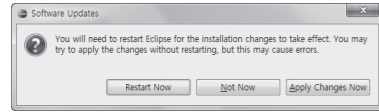
[그림 6-6] Install 창의 라이선스 동의 화면

07. 설치 과정에서 Security Warning 창이 뜰 수도 있는데 이 경우에는 OK 버튼을 클릭해서 계속 설치를 진행한다.



[그림 6-7] 보안 경고 창

08. 끝으로 Restart Now 버튼을 클릭해서 이클립스를 재시작하면 된다.

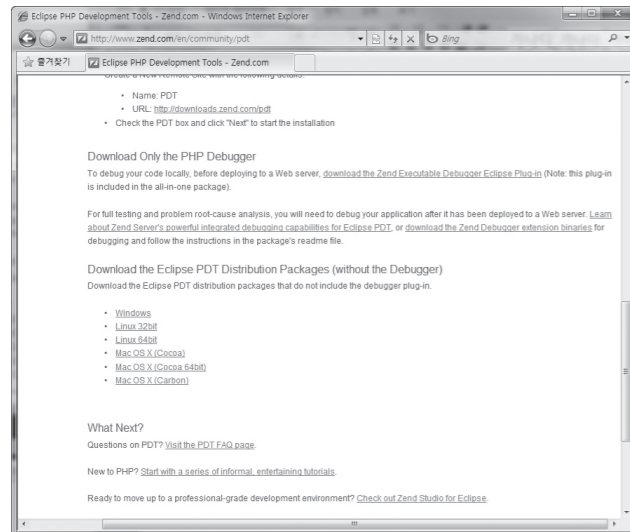


[그림 6-8] 이클립스 재시작 여부 확인

다운로드 설치 방법

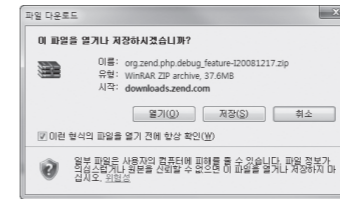
업데이트 사이트를 통해 설치하는 데 시간이 오래 걸리거나 그 밖의 이유로 설치하기가 여의치 않다면 웹 브라우저를 이용해서 Zend Debugger 플러그인을 직접 내려 받아 설치해보기 바란다. 내려 받은 Zend Debugger 파일은 압축 파일 형태이므로 이클립스 PDT를 설치할 때와 마찬가지로 파일의 압축을 해제한 후 이클립스가 설치된 폴더에 덮어쓰기만 하면 손쉽게 설치가 완료된다. Zend Debugger 파일을 내려 받아 설치하는 방법은 다음과 같다.

01. 웹 브라우저를 열고 Zend.com의 Eclipse PHP Development Tools 페이지(<http://www.zend.com/en/community/pdt>)에 접속한다.
02. 웹 페이지에 있는 Download Only the PHP Debugger 단락 가운데 download the Zend Executable Debugger Eclipse Plug-in 링크를 클릭한다.



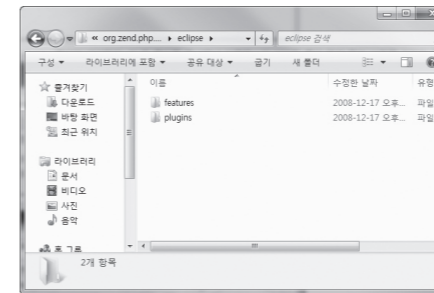
[그림 6-9] Zend.com Zend Debugger 다운로드 페이지

03. 웹 브라우저의 다운로드 창이 열리면 원하는 폴더에 Zend Debugger 플러그인을 저장한다.



[그림 6-10] Zend Debugger 다운로드 창

04. 내려 받은 Zend Debugger 플러그인 파일의 압축을 풀면 features와 plugins라는 폴더가 생성된다. 이 두 폴더에 들어 있는 파일을 이클립스 폴더에 있는 features와 plugins 폴더에 복사한다.



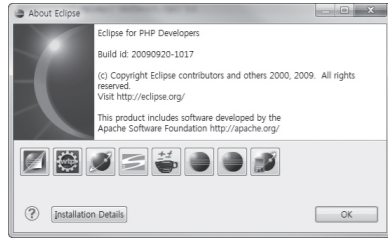
[그림 6-11] 압축 해제한 Zend Debugger 플러그인 폴더

05. Zend Debugger 플러그인 파일의 복사가 완료되면 이클립스를 다시 실행한다.

Zend Debugger 설치 확인

이클립스를 재시작한 후 Zend Debugger가 정상적으로 설치돼 있는지 확인하는 방법은 다음과 같다.

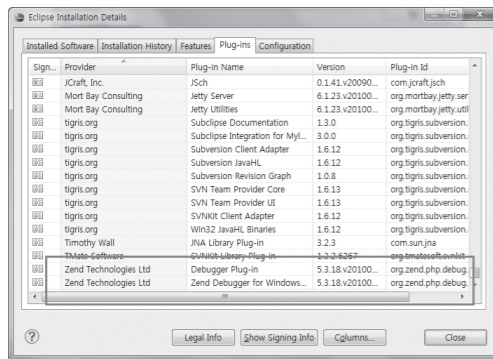
1. 이클립스 상단 메뉴에서 Help → About Eclipse를 선택한다.
2. About Eclipse 창이 열리면 Installation Details 버튼을 클릭한다.



[그림 6-12] About Eclipse 창

3. Eclipse Installation Details 창이 열리면 Plug-ins 탭을 클릭한다.
4. 이클립스에 설치된 플러그인 목록이 나타나면 목록 상단의 Provider를 클릭해서 Provider 순으로 정렬한다.
5. 그리고 Provider가 Zend Technologies Ltd이고 Plug-In Name이 아래 두 항목에 해당하는 것이 있는지 확인한다.

- Debugger Plug-in
- Zend Debugger for Windows



[그림 6-13] 이클립스에 설치된 플러그인 목록

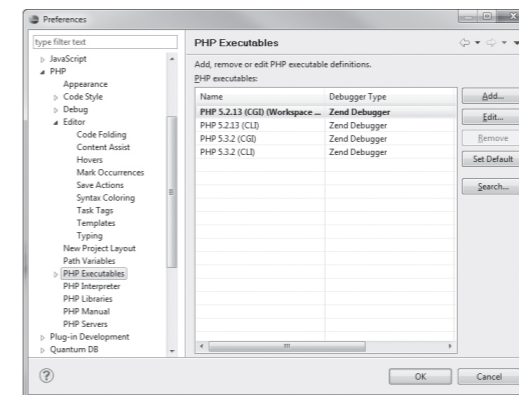
항목이 존재하지 않으면 위의 “Zend Debugger 설치”에서 살펴본 방법 중 다른 방법으로 Zend Debugger를 설치해 본다. 예를 들어 업데이트 사이트를 이용한 설치가 실패했다면 다운로드해서 설치하는 방법을 시도해 본다.

6-2. PHP 스크립트 실행

PHP 프로그램을 실행하는 방법은 PHP 스크립트 실행과 PHP 웹 페이지 실행으로 나눌 수 있다. 이 장에서 살펴볼 PHP 스크립트 실행은 PHP 프로그램을 콘솔에서 실행하는 것과 비슷하다. 콘솔에서 직접 PHP 프로그램을 실행하면 실행 결과가 웹 서버를 통해 출력되지 않고 콘솔의 표준 출력(STDOUT)으로 출력되듯이 이클립스 PDT의 PHP 스크립트 실행으로 PHP 프로그램을 실행하면 웹 브라우저로 출력돼야 할 실행 결과가 Debug Output 뷰를 통해 텍스트 형태로 출력된다. 이러한 특징은 셸 스크립트나 서버 프로그램과 같이 웹 브라우저로 출력할 필요가 없는 독립적인 PHP 프로그램을 개발할 때 유용하다. 또한 HTML 형식의 웹 문서를 출력하는 액션(Action) 로직과는 다르게 명확한 반환값을 출력하는 DAO(Data Access Object)나 BO(Business Object) 컨트롤러 로직을 개발할 때도 매우 유용하다.

PHP 스크립트 실행을 살펴보기 전에 다시 한번 강조하지만 이클립스에서 PHP 프로그램을 실행하려면 PHP 디버거가 꼭 설치돼 있어야 한다. PHP 디버거를 설치하지 않았다면 “Zend Debugger 플러그인 설치” 장에 자세히 설명돼 있으므로 참고하기 바란다.

Zend Debugger를 설치하면 이클립스 상단 메뉴에서 Window → Preferences를 클릭한 후 PHP → PHP Executables를 선택했을 때 [그림 6-14]와 같이 PHP 디버거가 여러 개 등록돼 있을 것이다. 이 중에서 굵은 글씨로 Workspace Default라고 표시된 항목이 PHP 프로그램을 실행할 때 기본적으로 사용하는 PHP 디버거이며, Debugger Type은 PHP 인터프리터가 어떤 디버거를 사용하는지를 나타낸다.

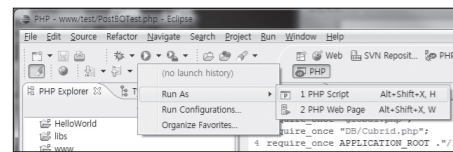


[그림 6-14] PHP Executables 화면

PHP 디버거의 설정을 변경하는 방법은 변경하고자 하는 PHP 인터프리터 항목을 선택하고 PHP Executables 화면 오른쪽에 있는 Edit 버튼을 클릭하면 된다. 그러나 이클립스 플러그인으로 설치한 경우에는 사실상 PHP 디버거의 설정을 변경할 수 없다. 그리고 다른 PHP 디버거를 기본(Default) 디버거로 지정하고 싶다면 지정할 PHP 디버거를 선택한 후 화면 오른쪽에 있는 Set Default 버튼을 클릭하면 된다.

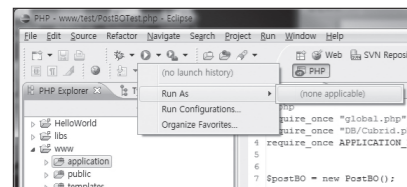
자, 이제 이클립스 PDT에서 PHP 스크립트를 실행하는 방법을 살펴보자.

01. 실행하고자 하는 PHP 파일을 편집기에서 연다.
02. 이클립스 PDT 상단 툴 바의 Run 버튼(🏃) 오른쪽에 있는 삼각형을 클릭한다. 편집기에서 아무런 PHP 파일을 열지 않은 상태에서 Run 버튼의 삼각형 버튼을 클릭하면 PHP Script 메뉴가 보이지 않을 것이다.
03. 메뉴에서 Run As → PHP Script를 선택한다.



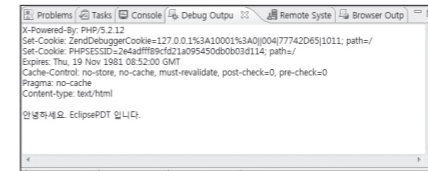
[그림 6-15] PHP 스크립트 실행

편집기에서 PHP 파일을 연 상태에서만 PHP Script 메뉴를 볼 수 있다는 점에 주의하자. PHP Explorer에서 폴더를 선택했거나 PHP 파일이 아닌 다른 파일을 편집기에서 활성화한 상태라면 다음 화면과 같이 PHP 실행 메뉴가 나타나지 않는다.



[그림 6-16] Run As 메뉴에 PHP Script 메뉴 항목이 없음

04. PHP 스크립트가 실행되면 Debug Output 뷰에 실행 결과가 출력된다.



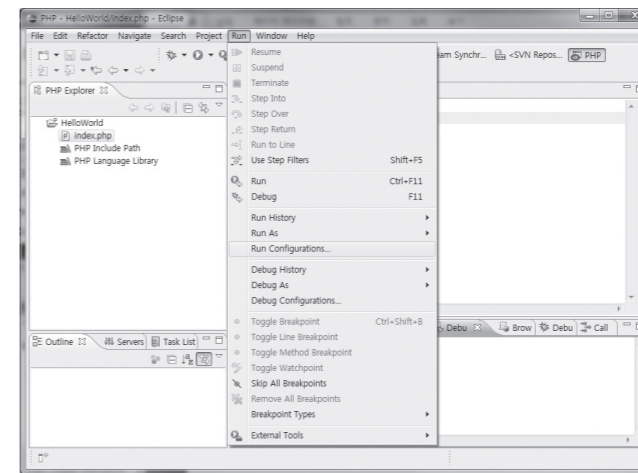
[그림 6-17] PHP 스크립트 실행 결과가 출력된 Debug Output 뷰

Debug Output 뷰에 출력된 실행 결과를 보면 특이하게도 X-Powered-By나 Set-Cookie, Content-type 같은 HTTP 헤더 값이 포함돼 있다. Debug Output 뷰에 HTTP 헤더 값이 출력되는 이유는 앞서 설명한 바와 같이 웹 서버를 통해 출력돼야 할 PHP 프로그램의 실행 결과가 Debug Output 뷰를 통해 그대로 출력되기 때문이다. 이러한 특징 덕분에 PHP 프로그램에서 setcookie() 함수나 header() 함수를 사용해 임의의 HTTP 헤더를 전송하고자 하는 경우 Debug Output 뷰에서 그러한 내용을 확인할 수 있다. 단, 웹 서버에서 덧붙이는 HTTP 헤더 값은 Debug Output 뷰에 출력되지 않는다.

PHP 실행 설정 변경하기

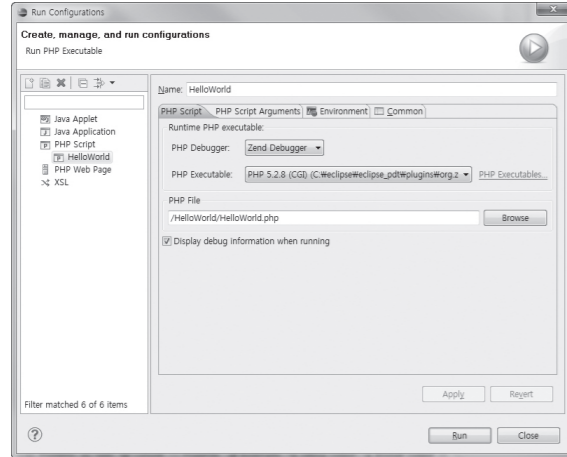
PHP 스크립트를 실행하면 실행과 관련된 설정 내용이 자동으로 저장되며, 다음과 같은 방법으로 설정 값을 변경하거나 삭제할 수 있다.

01. 이클립스 상단 메뉴에서 Run → Run Configurations를 선택한다.



[그림 6-18] PHP 실행 메뉴

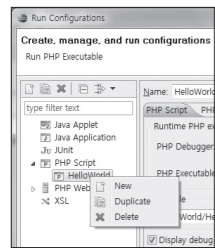
02. PHP Script 항목을 열면 실행했던 PHP 파일에 대한 설정 내용을 볼 수 있다. 여기서는 index.php 파일을 실행했으므로 index라는 항목이 저장돼 있다. index 항목을 선택하면 PHP Script 탭에서 실행에 관련된 내용을 확인할 수 있다.



[그림 6-19] PHP 실행 설정 화면

- 03. 실행할 대상 PHP 파일을 변경하려면 PHP File 입력란의 파일 경로를 원하는 PHP 파일로 변경한다.
- 04. 변경 내용을 저장하기 위해 활성화된 Apply 버튼을 클릭한다.
- 05. Run 버튼을 클릭하면 변경한 실행 설정으로 PHP 파일이 실행된다. 변경한 내용을 되돌리고 싶다면 Revert 버튼을 클릭한다.

PHP Script 설정을 복사하거나 삭제할 수도 있다. PHP Script 항목에서 삭제하고자 하는 설정 항목을 선택하고 마우스 오른쪽 버튼을 클릭한다. 그리고 나서 [그림 6-20]과 같이 선택 메뉴가 나타나면 Delete를 선택한다.



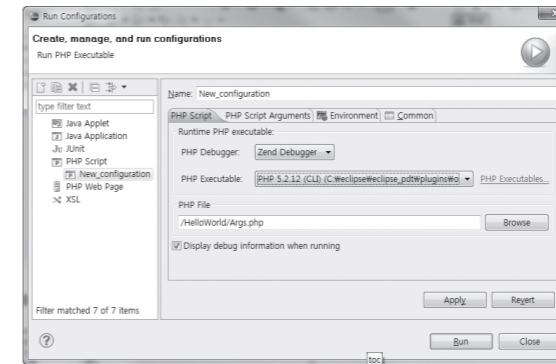
[그림 6-20] Run 설정 항목 삭제

PHP 스크립트의 인자 값 전달

예를 들어 윈도우 콘솔에서 dir /a를 실행하거나 리눅스의 ls -la를 실행할 때처럼 콘솔에서 어떤 프로그램을 실행할 때 인자 값을 전달할 수 있다. 이클립스에서 PHP 스크립트를 실행할 때도 콘솔에서 프로그램에 인자 값을 전달하는 것과 같이 PHP 프로그램에 인자 값을 전달할 수 있다.

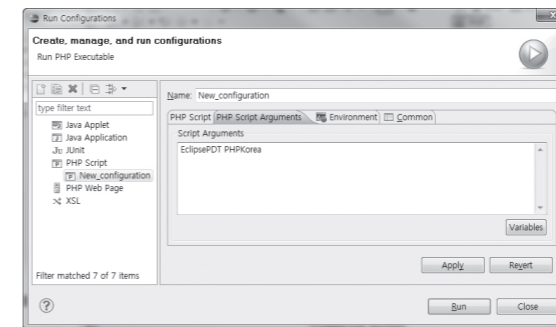
PHP 스크립트를 실행할 때 인자 값을 전달하는 방법은 다음과 같다.

- 01. 이클립스 상단 메뉴에서 Run → Run Configurations를 차례로 선택해서 PHP Script 실행 설정 화면을 연다.
- 02. PHP Script 항목에서 인자 값을 전달해서 실행할 PHP 실행 항목을 선택한다.



[그림 6-21] PHP 스크립트 실행 설정

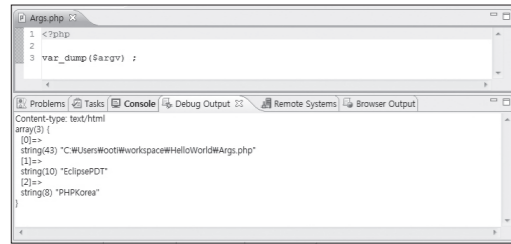
- 03. PHP Script Arguments 탭을 클릭한 후 전달하고자 하는 인자 값을 입력한다.



[그림 6-22] PHP 스크립트 인자 값 입력

04. Run Configurations 창 하단에 있는 Run 버튼을 클릭한다.

05. PHP Script Arguments 에 입력한 값은 \$argv라는 배열로 PHP 프로그램에 전달된다.



[그림 6-23] PHP 스크립트 인자 값 확인

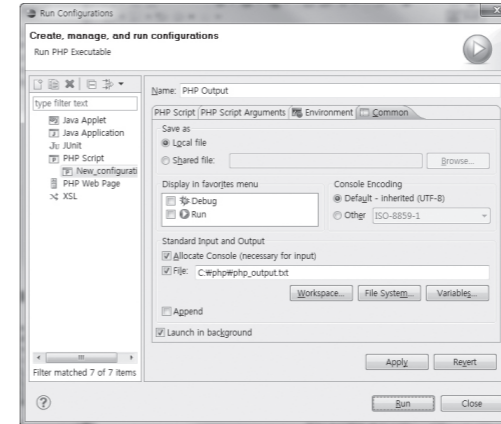
\$argv에 저장된 실행 인자 값은 PHP 소스 코드에 다음과 같이 입력한 후 실행해 보면 확인할 수 있다.

```
var_dump($argv);
// 또는
print_r($argv);
```

실행 결과를 파일로 저장하기

PHP 스크립트를 실행한 후의 결과 값이 복잡하고 많거나 결과 값을 지속적으로 기록해야 하는 경우 실행 결과를 파일로 저장하면 좋다.

PHP 스크립트 실행 결과를 파일로 저장하려면 앞서 살펴본 PHP 스크립트 실행 설정 화면에서 Common 탭을 클릭한다. 그리고 나서 [그림 6-24]와 같이 File 입력란을 체크하고 출력 내용이 저장될 파일 경로를 입력하면 된다. 이제 Run 버튼을 클릭하면 PHP 스크립트가 실행되고 실행 결과가 File 입력란에 입력한 파일에 텍스트 형태로 저장될 것이다.



[그림 6-24] PHP 스크립트 실행 설정 화면의 Common 탭

그런데 실행 결과를 파일에 저장하면 PHP 스크립트를 실행할 때마다 해당 파일에 실행 결과를 그대로 쓰기 때문에 기존에 저장돼 있던 실행 결과가 사라진다. 이전 실행 결과를 지우지 않고 새로운 실행 결과를 저장하고 싶다면 [그림 6-24]의 File 입력란 아래에 있는 Append를 체크하면 된다. 그러면 PHP 로그 파일처럼 기존 내용이 지워지지 않고 새로운 내용이 덧붙여 기록된다.

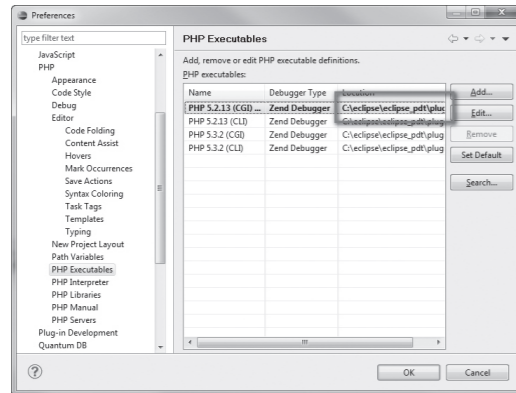
PHP 인터프리터의 버전 변경

Zend사에서 이클립스용 Zend Debugger 플러그인을 개발, 배포하고 있긴 하지만 php.net에 공개된 최신 버전의 PHP와는 다소 차이가 있을 수 있다. 약간의 버전 차이가 PHP 프로그램을 개발하는 데 큰 문제를 야기하지는 않지만 개발 환경의 PHP 버전과 실제 웹 서버의 PHP 버전이 다르면 왠지 불안하기도 하고 PHP 버전 차이로 인한 버그를 모른 채 지나칠 수도 있다.

그러므로 Zend사에서 공식적으로 배포하는 이클립스용 Zend Debugger 플러그인 버전과는 상관없이 사용자가 직접 Zend Debugger 플러그인에 포함돼 있는 PHP 인터프리터 버전을 변경해야 할 경우가 있으며, 이 경우 다음과 같은 방법으로 이클립스용 Zend Debugger 플러그인에 포함된 PHP 인터프리터 버전을 변경할 수 있다.

- 우선 PHP 인터프리터 플러그인이 설치된 경로를 확인해야 한다. 이클립스의 Preferences 창을 열고 PHP → PHP Executables를 선택하면 다음 화면과 같이 Location 항목을 볼 수 있는데 이 경로가 이클립스용 Zend Debugger 플러그인에 포함돼 있는 PHP 인터프리터 경로에 해당한다.

예) C:\eclipse\eclipse_pdt\plugins\org.zend.php.debug.debugger.win32.x86_5.2.15.v20081217\resources



[그림 6-25] PHP 디버거 설정 창

- PHP 인터프리터가 설치돼 있는 폴더를 열면 PHP를 실행하는 데 필요한 최소한의 파일들을 볼 수 있다. 이 중에서 ZendDebugger.dll 파일을 다른 폴더에 임시로 복사해 둔다.
- PHP.net의 다운로드 페이지에서 내려 받은 최신 버전의 PHP 파일의 압축을 풀어 PHP 인터프리터가 설치된 폴더에 덮어쓴다.
- 임시 폴더에 복사해 둔 ZendDebugger.dll 파일을 다시 PHP 인터프리터의 플러그인 폴더에 복사해 넣는다.
- php.ini 파일을 열고 extension_dir을 찾아 다음과 같이 변경한다.

```
extension_dir = "./ext/"
```

- 마지막으로 php.ini 파일 내용의 extension 부분에 다음 코드를 추가한다.

```
zend_extension_ts=.\\ZendDebugger.dll
```

- 앞으로 이클립스에서 PHP 스크립트를 실행하면 변경한 PHP 버전에서 PHP 프로그램이 실행된다.

Zend Debugger 플러그인에 모듈 추가하기

이클립스용 Zend Debugger 플러그인의 PHP 인터프리터 버전을 변경한 것과 마찬가지로 Zend Debugger 플러그인에 포함된 PHP Extension을 추가하거나 변경할 수 있다. PHP Extension을 변경하는 방법은 다음과 같다.

- 이클립스용 Zend Debugger 플러그인의 PHP 인터프리터가 설치된 폴더를 연다.
- ext 폴더에 추가하고자 하는 PHP Extension을 복사해 넣는다.
- php.ini 파일에 extension을 추가한다.

6-3. PHP 웹 페이지 실행

이클립스 PDT에서는 PHP 웹 페이지 실행이라는 기능을 제공한다. PHP 웹 페이지 실행은 앞서 살펴본 PHP 스크립트 실행과는 방법에 있어서 큰 차이가 있다. PHP 스크립트 실행이 이클립스용 PHP 디버거를 설치해서 PHP 프로그램을 실행하는 반면 PHP 웹 페이지 실행은 웹 서버를 통해 PHP 프로그램을 실행하고 그 결과를 이클립스 PDT의 PHP Browser로 출력한다. 다시 말해서 PHP 스크립트 실행이 콘솔에서 실행하는 것과 같다면 웹 페이지 실행은 웹 서버와 연동해 실제 서비스에서 PHP 프로그램을 구동하듯 실행하는 것이다. 이러한 PHP 웹 페이지 실행은 개발 과정에서 PHP 프로그램의 액션(Action) 로직을 실행하는 데 유용하다.

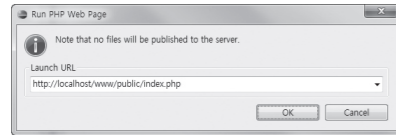
PHP 웹 페이지 실행을 위해서는 PHP가 설치된 웹 서버가 필요하다. 일반적으로 이클립스가 설치된 로컬 PC나 네트워크를 통해 접속할 수 있는 컴퓨터에 웹 서버를 설치해서 사용한다. 필자는 이클립스를 사용해야 하는 로컬 PC보다는 별도의 개발용 웹 서버 PC를 하나 더 마련해서 사용했다. 웹 서버가 아무리 가벼워도 개발용 PC의 자원을 사용하는 것은 사실이며 웹 프로그램의 특성상 DBMS도 함께 설치해야 하는 경우가 많기 때문이다.

PHP 웹 페이지 실행을 위한 절차는 다음과 같다.

- 웹 페이지 실행에 필요한 파일을 모두 웹 서버로 전송한다.
- PHP 파일에 접근 가능하게끔 웹 서버를 구동한다.
- 이클립스 PDT의 PHP Explorer에서 실행하고자 하는 파일을 선택하고 마우스 오른쪽 버튼을 클릭한다.

04. 문맥 메뉴가 나타나면 Run As → PHP Web Page를 선택한다.

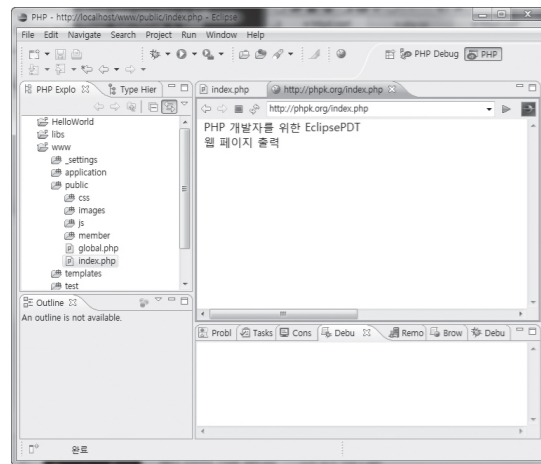
05. Run PHP Page 창이 열리면 Launch URL에 실행하고자 선택했던 파일에 대한 웹 서버 URL을 입력한다. 예를 들어 실행하고자 하는 PHP 파일명이 index.php이고 이 파일에 대한 웹 서버 URL이 http://localhost/www/public/index.php라면 Launch URL에 이 URL을 입력하면 된다



[그림 6-26] PHP Web Page의 Launch URL 선택 창

웹 서버를 이클립스가 설치된 로컬 PC에 설치해서 사용하고 있다면 웹 서버의 DocumentRoot를 이클립스 PDT의 Workspace로 지정할 것을 권장한다. 왜냐하면 이클립스 PDT는 Workspacer가 DocumentRoot로 지정돼 있다는 것을 전제로 해서 PHP 웹 페이지 실행을 하기 때문이다. 단, 실제 서비스의 URL과는 차이가 있다는 점을 고려해서 프로그램을 설계해야 한다.

06. 이클립스 안에 PHP 웹 브라우저가 열리면서 웹 페이지가 실행된다.



[그림 6-27] PHP 웹 페이지 실행 결과

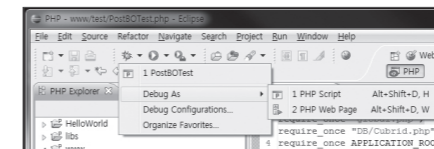
6-4. PHP 스크립트 디버깅

오래 전 필자가 PHP 프로그램을 개발할 때는 디버깅이라 하면 소스 코드 사이에 echo 문이나 printf() 함수를 집어넣는 방법밖에 없었다. 일명 디버깅 코드라고 하는 임시 코드를 정상적인 소스 코드 사이에 끼워 넣은 후 프로그램을 실행하는 것이다. 그리고 나서 개발자가 의도한 대로 변수 값이 잘 전달되고 있는지, 어떤 값이 입력돼 있는지, 결과는 잘 출력되는지 등을 확인했다. 디버깅이 끝나면 소스 코드에서 디버깅 코드를 지우는 작업을 해야 했다. 디버깅 코드를 사용해서 디버깅하는 방법은 많은 프로그래밍 언어에서 일반적으로 사용하는 디버깅 방법이라고 생각한다. 특히 간단하게 디버깅하는 경우에는 아주 유용한 방법이지만 가끔 필자에게는 이 방법이 매우 불편하게 느껴졌다. 예를 들어 프로그램이 실행되는 동안 특정 변수 값의 변화를 확인하고 싶다고 가정하자. 디버깅 코드를 프로그램 전체 소스 코드에 걸쳐 추가할 수도 없고 이렇게 한다고 해서 값이 변화하는 시점을 개발자가 정확하게 인식할 수 있다고 보장할 수도 없다. 그렇기 때문에 여러 차례에 걸쳐 디버깅 코드를 삽입·제거하는 과정을 반복하게 될 것이다. 더불어 디버깅 코드를 깨끗하게 제거하지 않은 채 실제 서비스 서버에 배포하면 경우에 따라서는 개인정보 유출 같은 심각한 문제가 초래될 수도 있다.

이클립스 PDT에서는 향상된 PHP 프로그램의 디버깅 방법을 제공한다. 단, PHP 스크립트 디버깅을 하려면 반드시 이클립스 PDT에 PHP 디버거가 설치돼 있어야 한다. PHP 디버거를 설치하는 방법은 “Zend Debugger 설치”에서 자세히 설명하고 있으므로 먼저 읽어보기 바란다.

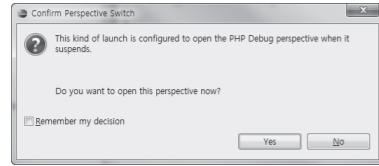
이클립스 PDT에서 PHP 스크립트를 디버깅하는 방법은 다음과 같다.

01. 디버깅하고 싶은 파일을 편집기에서 연다.
02. 이클립스 상단 툴 바에서 벌레 버튼을 클릭한다.
03. 메뉴에서 Debug As → PHP Script를 선택한다. 기존에 실행했던 PHP 스크립트 설정의 이름을 클릭해도 된다. 한번이라도 실행 또는 디버깅한 PHP 스크립트 설정은 실행이나 디버그 메뉴에 등록되어 표시된다.



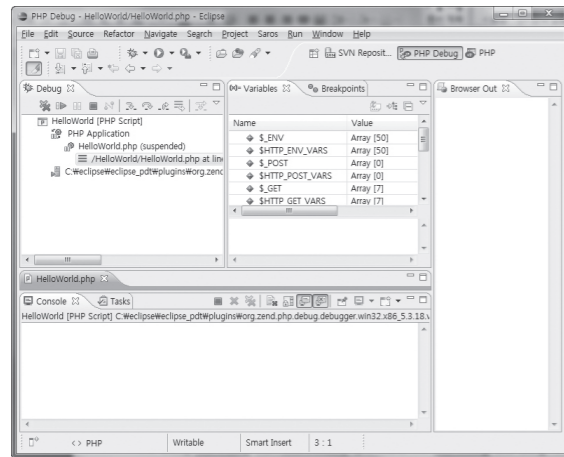
[그림 6-28] PHP 디버깅 메뉴

04. 퍼스펙티브를 변경할 것인지 묻는 창이 나타나면 Yes를 클릭한다.



[그림 6-29] PHP Debug 퍼스펙티브로 변경 확인

05. 이클립스가 PHP Debug 퍼스펙티브로 변경된다.

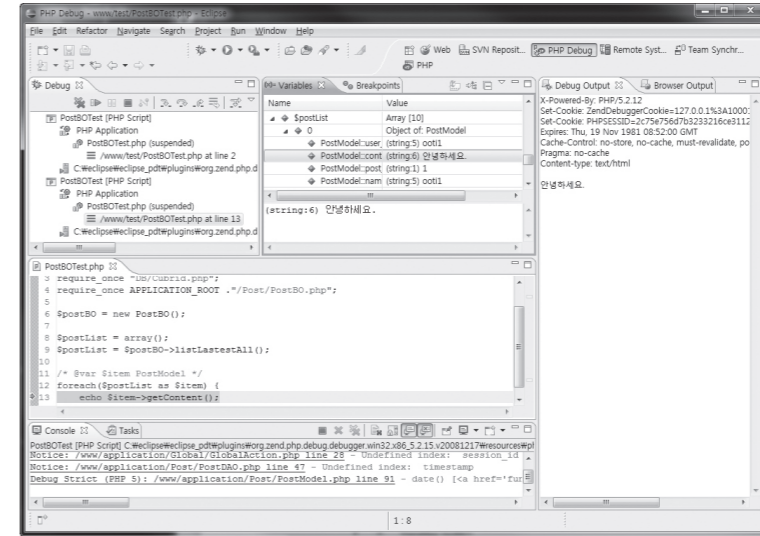


[그림 6-30] PHP Debug 퍼스펙티브

06. **F6** 키를 누르면 PHP 프로그램이 실행되는 순서대로 PHP 디버깅이 수행된다. 디버깅을 종료하는 방법은 **Ctrl** + **F2** 키를 입력하거나 Console 뷰의 상단 톨바에 있는 빨간색 Terminate 버튼을 클릭하면 된다.

PHP Debug 퍼스펙티브

PHP Debug 퍼스펙티브는 PHP 퍼스펙티브와는 많은 차이가 있다. PHP 퍼스펙티브는 소스 코드를 작성하기 위해 편집기 중심으로 구성돼 있었지만 PHP Debug 퍼스펙티브는 PHP 프로그램을 분석하기 위한 Debug 뷰와 Variables 뷰를 중심으로 배치돼 있다. 또한 PHP Debug 퍼스펙티브에서 제공되는 편집기는 소스 코드의 어느 부분이 진행 중인지 보여주는 용도로 사용된다.



[그림 6-31] PHP 스크립트 디버그 화면

이클립스 PDT 상단에 위치한 Debug 뷰는 현재 어떤 PHP 인터프리터를 사용하고 있고 어떤 PHP 파일의 몇 번째 줄에 대한 디버깅을 진행하고 있는지 등 중요한 디버깅 정보를 트리 형식으로 보여 준다. Debug 뷰에 나타난 항목 가운데 다음 화면과 같이 line이 표시된 항목을 클릭하면 Variables 뷰를 통해 현재 실행 중인 줄의 변수 목록을 확인할 수 있다.

///www/test/PostBOTest.php at line 13

Variables 뷰에서 원하는 항목을 클릭하면 바로 아래 빈 영역에 선택한 변수에 할당된 값이 나타나고 편집기에는 현재 진행 중인 소스 코드가 표시된다. Debug Output 뷰는 이클립스 화면 오른쪽에 있고 이클립스 화면 아래쪽에는 Console 뷰가 있다. Console 뷰에는 Debug Output 뷰로는 출력되지 않는 내부 경고나 오류 메시지 등이 출력된다.

디버깅 컨트롤

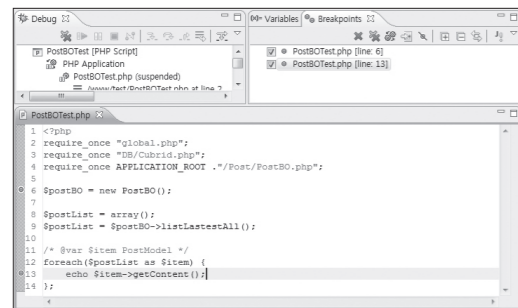
- 디버깅 중단
- 브레이크 포인트가 나타날 때까지 계속 실행(F8)
- 일시 정지

- ■ 실행 중단(Ctrl) + (F2)
- 🔄 연결 끊기
- 🏠 함수나 메서드 내부로 진입(F5)
- 🏠 함수나 메서드 내부로 진입하지 않고 다음을 실행(F6)
- 🏠 현재 진입한 함수나 메서드를 빠져나가 다음을 실행(F7)

브레이크 포인트 지정

브레이크 포인트는 디버깅 과정에서 멈춰서는 지점을 말한다. 예를 들어 1000줄의 소스 코드가 있고, 이 중에서 100번째 줄부터 디버깅을 시작하고 싶다면 1번째 줄부터 99번째 줄까지는 디버깅하지 않고 건너뛰어야 할 것이다. 만약 그렇게 하지 않는다면 100번째 줄에 다다를 때까지 사용자는 수십 번에 걸쳐 진행 버튼(F6) 키를 눌러야 할 수도 있다. 일반적으로 Debug 버튼을 클릭하기 전에 미리 브레이크 포인트를 지정한다. 디버깅이 시작되면 디버거는 첫 번째 브레이크 포인트에서 자동으로 일시 정지한다. 이는 DVD를 보다가 일시 정지(Pause) 버튼을 누른 것과 같다. 디버거는 멈춰선 지점에서의 실행 정보를 Debug 뷰를 통해 자세히 보여준다.

브레이크 포인트를 지정하는 것은 아주 쉽다. 이클립스 PDT의 편집기에 브레이크 포인트를 걸고 싶은 PHP 파일을 연다. 그리고 나서 편집기 맨 왼쪽의 줄 번호 바로 앞의 빈 영역을 마우스로 더블 클릭한다. 그러면 [그림 6-32]와 같이 자그마한 원이 클릭한 위치에 나타나며, Breakpoints 뷰에 파일명과 줄 번호가 표시된다. Breakpoints 뷰에서 항목을 선택하고 마우스로 더블 클릭하면 편집기를 통해 소스 코드를 확인할 수 있다.



[그림 6-32] Breakpoints 뷰

이제 앞에서 살펴본 PHP 스크립트 디버깅 절차에 따라 Debug를 실행한다. 한 가지 주의할 점은 PHP 스크립트 디버깅의 기본 설정은 PHP 프로그램의 맨 첫 번째 줄에서 자동으로 일시 정지하도록 돼 있어서 브레이크 포인트가 지정돼 있지 않더라도 맨 첫 번째 줄에서 무조건 일시 정지한다는 것이다. 그 상태에서 F8 키를 누르거나 ▶ 버튼을 클릭하면 브레이크 포인트로 지정한 첫 번째 위치로 이동한다.

6-5. 로컬 웹 서버를 연동한 디버깅

궁극적으로 PHP 프로그램은 웹 서버에서 동작할 프로그램이므로 개발자는 작성한 PHP 프로그램을 실제 웹 서버에 반영하기 전에 개발용 웹 서버에서 테스트할 수밖에 없다. 웹 프로그램을 개발할 때 자주 쓰는 방법은 로컬 PC에 웹 서버를 설치해서 테스트하는 것이다. 이렇게 하면 이클립스 같은 개발도구에서 프로그램을 개발한 뒤 배포 과정을 거치지 않고도 빠르게 결과를 확인할 수 있기 때문이다. 이클립스 PDT와 Zend Debugger는 웹 서버와 연동해서 디버깅하는 방법을 제공한다.