

Hanbit eBook

Realtime 20

프로그래머를 위한 가이드

드루팔

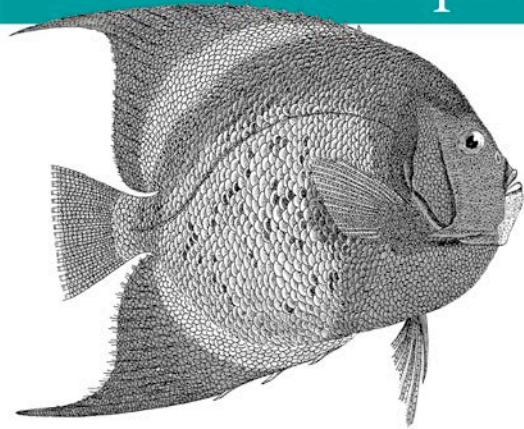
Programmer's Guide to Drupal

제니퍼 하지던 지음 / 김지원 옮김

O'REILLY®  한빛미디어
Hanbit Media, Inc.

Principles, Practices, and Pitfalls

Programmer's Guide to Drupal



O'REILLY®

Jennifer Hodgdon

이 도서는 O'REILLY의
Programmer's Guide to Drupal의
번역서입니다.

프로그래머를 위한 가이드
드루팔

프로그래머를 위한 가이드 **드루팔**

초판발행 2013년 4월 30일

지은이 제니퍼 하지던 / **옮긴이** 김지원 / **펴낸이** 김태헌
펴낸곳 한빛미디어(주) / **주소** 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부
전화 02-325-5544 / **팩스** 02-336-7124
등록 1999년 6월 24일 제10-1779호
ISBN 978-89-6848-608-1 15000 / **정가** 11,000원

책임편집 배용석 / **기획·편집** 김창수
디자인 표지 여동일, 내지 스튜디오 [ميم], 조판 북누리
마케팅 박상용, 박주훈

이 책에 대한 의견이나 오타자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주세요.

한빛미디어 홈페이지 www.hanb.co.kr / **이메일** ask@hanb.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2013 HANBIT Media, Inc.

Authorized Korean translation of the English edition of *Programmer's Guide to Drupal*, ISBN 9781449343316

© 2013 Polar Productivityware, LLC. This translation is published and sold by permission of O'Reilly Media, Inc., which owns or controls all rights to publish and sell the same.

이 책의 저작권은 오라일리사와 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanb.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

저자 소개

지은이_ **제니퍼 하지던** Jennifer Hodgdon

제니퍼 하지던은 1982년에 처음으로 자신의 컴퓨터 프로그램을 제작했다. 그러다가 1994년에 전문 소프트웨어 개발자의 길로 발을 들였는데, 당시 제니퍼는 포트란, C, 자바, PHP, 자바스크립트 등 다양한 프로그래밍 언어를 구사했다. 제니퍼는 2002년경 PHP와 MySQL을 이용한 웹 프로그래밍을 본업으로 시작했고, 2007년엔 드루팔로 자신의 첫 홈페이지를 만들었다. 그로부터 얼마 지나지 않아, 드루팔 오픈 소스 프로젝트와 시애틀 드루팔 사용자 그룹에 기여하는 데 시간을 투자하기 시작했다. 모임과 콘퍼런스를 준비하고, 2011년부터 2012년까지는 공동 대표로 활동하다가 드루팔 문서화 팀장을 맡았고, 워크숍을 주최하고 콘퍼런스 발표자로 참여했다. 제니퍼는 현재 프리랜서로 드루팔 사이트를 제작하고 드루팔 모듈을 개발한다. 그리고 무보수로 드루팔 모듈 유지보수자, 스포캔Spokane, 워싱턴 주 드루팔 사용자 그룹의 공동주최자, API 문서화 및 코딩 표준을 위한 드루팔 핵심 유지보수자와 커미터로 활동하며 커뮤니티에 기여하고 있다. 제니퍼에게 연락할 일이 있다면 제니퍼의 비즈니스 사이트인 <http://www.poplarware.com> 이나 <http://www.drupal.org/user/155601> 페이지를 이용하면 된다.

역자 소개

역자_ 김지원

웹 기술뿐 아니라 온갖 분야에 발을 뻗고 싶어 하는 바람기를 지녔지만 역부족이다. 배워야 할 것이 갈수록 늘어나 시간의 결핍을 느낀다. 기술 문서, 매뉴얼, 유비쿼터스 관련 논문을 번역한 바 있고 해외 논문 DB를 구축 관련 작업에도 참여했다. 『한 권으로 끝내는 정규표현식』(2010), 『웹 표준 가이드: HTML5+CSS3』(2010), 『프로젝트로 배우는 HTML5+자바스크립트』(2012), 『리팩토링』(2012), 『Elemental Design Patterns』(2013, 이상 한빛미디어) 등을 번역했다.

저자 서문

이 글을 보는 여러분을 환영한다. 이 책은 드루팔^{Drupal}이라는 이름으로 유명한 오픈 소스 웹 콘텐츠 관리 시스템을 사용해 개발하고자 하는 여러분을 위해 만들었다. 이 책을 도우미로 두면, 여러분은 이제껏 수많은 드루팔 전문 개발자들이 드루팔 개발을 시도하면서 저질렀던 실수를 피해 가면서 초보 드루팔 프로그래머 단계를 순탄하고 매끄럽게 지나갈 수 있을 것이다. 이 책의 설명에 따라 ‘드루팔 방식^{Drupal Way}’으로 프로그래밍하는 법을 배우면, 틀림없이 드루팔 개발에 투자하는 시간이 다양한 즐거움과 성취감의 결실로 가득할 것이다.

저자
제니퍼 하지던

역자 서문

드루팔은 현재 국제적인 3대 주요 콘텐츠 관리 시스템인 워드프레스, 드루팔, 줌라^{Joomla} 중에서 가장 역사가 오래된 CMS로 국내에도 어느 정도 사용자층을 형성하고 있다.

드루팔의 창안자인 드리스 바이타르트^{Dries Buytaert}는 1999년 게시판을 드루팔을 사용해 개발을 시작했다. 그 후 1년간 많은 이들이 드루팔에 관심을 갖고 코드를 공유함으로써 오픈 소스 프로젝트로 자리매김했다. 특히 많은 CMS 중에서도 드루팔은 특히 모듈화 특성이 강조돼 있다. 사실 드루팔에 대한 전반적인 사항을 일일이 설명해주는 책을 기대하는 사람들에게겐 어쩌면 이 책은 적합하지 않을지도 모른다. 드루팔의 초보자이지만 개발 지식은 어느 정도 있고, 적어도 PHP나 MySQL 정도의 기초 지식은 가지고 있어야 하며 드루팔로 만든 자신의 사이트가 마음에 안 들어 일부 코드나마 수정해 본 사람으로서 핵심을 짚어주는 책이 필요한 사람에게 적합하다는 생각이 든다.

그만큼 이 책은 드루팔의 전반을 두루 훑고 있진 않다. 대신에, 중간 이하의 난이도임엔 분명하지만, 기본적인 사전 지식이 있으면서 이제 막 드루팔 설치를 끝냈지만 전체적 구조를 살펴보고 대략 감 잡은 사람이라면 이 책이 짚어주는 다소 깊이 있고 핵심적인 노하우를 캐치해 누구와도 공유할 수 있는 표준에 가까운 자신의 모듈이나 테마 등의 산물을 그 술한 시행착오를 건너뛴 채 만들어낼 수 있을 것이다. 아무래도 오픈 소스 프로젝트인 만큼 커뮤니티에서 도움을 얻고 또 자신도 그 커뮤니티에 기여를 하면서 드루팔을 이용한 사이트나 애플리케이션 개발에 전념한다면 더 빠른 발전이 있을 거라고 생각한다. 내가 만든 소프트웨어를 남들이 사용하고 있음을 알게 될 때의 기쁨은 아마 처음이 아닌 사람은 익히 알고 있을 것이다. 그 기쁨은 꾸준히 쌓여 자신의 지속적인 개발에 추진력을 발휘하게 해준다.

2장의 원칙들은 드루팔 개발을 하면서 저자들이 수년간의 시행착오를 겪으며 얻은 노하우를 유지보수가 용이한 드루팔 코드를 개발하는 데 필수적인 몇 가지 원칙으로 정리한 것이다. 그 원칙들만 지키면 드루팔의 버전이 업데이트되거나 자신의 코드를 기부해서 남들이 사용하게 됐을 때 표준 호환성이 높기에 거의 문제 없이 적용할 수 있다. 그리고 3장의 실수 역시 저자들의 시행착오를 통한 노하우로부터 어떻게 하면 그런 실수를 방지할 수 있는지 비결을 요약해 놓았다. 드루팔 개발자로서 거의 필수적으로 하게 되는 4장의 기본적인 실전 코딩을 따라해 보면서 2장과 3장의 내용을 명심하는 습관을 들이면, 자신이 계획하는 프로그래밍은 유연성과 유지보수성, 사용성이 높은 결과물로 탄생할 것이다.

끝으로, 이 책을 발굴하고 저에게 번역의 기회를 준 스마트미디어팀과 김창수 팀장님께 특별한 감사를 표한다. 언제나 저를 격려해 주고 저의 피곤해 보이는 얼굴에 아낌없이 위로를 보내준 한빛미디어의 모든 분들께도 감사를 표한다. 이 책을 읽고 많은 분들이 전 세계인에게 공유되는 체계적이고 멋진 드루팔 코드를 만들 수 있길 바란다.

대상 독자

초급

초중급

중급

중고급

고급

이 책은 드루팔 활용과 프로그래밍에 입문하는 이들을 위한 초급 서적이다. 기존 프로그래밍 지식이 있는 사람은 이 책을 읽고 자신의 지식을 드루팔 활용이나 개발에 최적으로 적용할 수 있다. 뿐만 아니라 원하는 결과를 얻기 위해서 자신의 프로그래밍 기술을 어디에 어떻게 적용할 것인지 아는 드루팔 전문 개발자로 거듭나게 된다.

- 내부가 어떤 원리로 돌아가는지 알고자 하는 드루팔 사용자
- 원하는 결과를 얻기 위해 드루팔 방식의 프로그래밍 기법을 배우려는 드루팔 사이트 제작자와 테마 디자이너
- 드루팔 오픈 소스 프로젝트에 프로그래머로 기여하고 싶은 사용자

드루팔의 서버 측 처리와 코드 대부분이 약간의 SQL 데이터베이스 질의 언어를 곁들인 PHP 스크립트 언어로 작성된다. 따라서 플래시나 자바스크립트를 비롯한 여러 다른 클라이언트 측 언어도 드루팔에 연동할 수 있지만, 이 책은 PHP 및 데이터베이스 프로그래밍을 중점으로 다룬다. 이 책은 드루팔 개발자를 대상으로 하므로, 독자는 웹과 HTTP 요청에 대한 기초 지식을 가지고 있어야 하며, PHP 언어의 기본과 일반적인 프로그래밍 개념을 알고 있어야 한다. 이 책에서는 일반적인 프로그래밍 용어를 설명하지 않는다.

이 책을 효과적으로 보는 법

이 책의 내용을 최대한 자기 것으로 만들려면 1장을 먼저 읽어서 기본적인 내용을 완전히 숙지해야 한다. 드루팔을 설치하거나 전혀 사용해 본 적이 없다면 지금 당장 설치하자. <http://drupal.org/documentation/install>에서 드루팔 배포판을 다운받으면 그 안에 설치 방법이 수록된 INSTALL.txt 파일이 들어 있다.

설치했다면 드루팔 프로그래밍 샘플 코드를 볼 수 있게끔 <http://drupal.org/project/examples>에서 Examples for Developers 프로젝트를 받아야 한다. 여기엔 드루팔 코어(애드온을 제외한 드루팔의 기본 시스템)와 더불어 다양한 드루팔 샘플 코드가 들어 있다. 샘플 프로젝트는 드루팔 커뮤니티 소속의 여러 코드 기부자들이 수정 보완한 결과물로, 아주 좋은 참고 자료로 사용할 수 있다. 1장은 배경 지식 전달에만 충실하므로 그런 샘플 코드를 보면서 1장을 읽어 내려가면 어느덧 그런 샘플 프로젝트 수준을 뛰어넘어 드루팔 전문 개발자가 돼 있을 것이다.

샘플 프로젝트도 살펴봤으면 그 프로젝트의 샘플 모듈을 한두 개 설치해서 사용해보고 코드를 훑어보자. 포함된 프로젝트 모듈의 설치법도 역시 README.txt 파일에 들어 있다. 유난히 끌리는 주제가 있다면 그 주제와 관련된 모듈을 택하면 된다. Block 샘플 모듈과 Page 샘플 모듈이 기본에 충실하므로 추천한다. 샘플 모듈 코드를 가지고 공부할 때, 특정 드루팔 API 함수의 설명을 보고 싶으면 드루팔 API 매뉴얼 공식 사이트인 <http://api.drupal.org>를 참고하자.

이렇게만 해도 드루팔 코드를 보는 법을 어느 정도는 감 잡을 수 있다. 이 시점에서 2장과 3장을 차근차근 읽으면 드루팔 프로그래밍을 할 때 지켜야 할 원칙과 하지 말아야 할 실수가 무엇인지 알 수 있다.

그런 다음 특정 주제와 4장에 있는 실전 코드를 직접 작성해볼 차례인데, 이를 위해서는 배경 지식이 필요하므로 다시 Examples for Developers 샘플 프로젝트에서 관심 가는 프로젝트 코드를 살펴보자. 그 코드를 빠르게 훑어봐서 어떤 것들이 들었는지 알았으면 필요할 때마다 이 책으로 돌아와 각각에 해당하는 주제와 예제를 보자.

끝으로 5장의 팁과 조언을 읽어보자. 이 책의 여러 절에는 그 뒤에 설명할 내용을 위한 조언이 들어 있다.

예제 테스트 환경

드루팔 버전

몇 년마다 드루팔 프로젝트는 드루팔 6이나 7 같은 드루팔의 새 메이저 버전을 출시한다. 메이저 버전이 업그레이드될 때마다 호환성이 없어지고 구조나 API에 큰 변화가 생긴다. 그래서 보통은 한 메이저 버전을 이용해 개발한 코드는 다른 메이저 버전에서 수정하지 않은 채 사용할 수가 없다. 커뮤니티 개발자들이 기부한 모듈(drupal.org에서 받을 수 있다) 역시 Views 모듈 6.x-2.x와 6.x-3.x 같이 크게 달라져서 구조와 API가 다른 메이저 버전에서 호환되지 않는다.

이 책의 샘플 코드는 드루팔 7과 기부 모듈 드루팔 7 버전대에 호환되며, 호환성 여부는 각 절에 언급했다. 이 책의 설명도 대부분 드루팔 7을 위주로 작성한 것이며, 드루팔 8에서 예상되는 변동사항은 '참고' 박스 안에 설명했다. 드루팔 8 버전은 이 책을 번역하는 현시점에 개발이 진행 중에 있다. 드루팔 홈페이지에 나와 있는 장기 계획표대로라면 2013년 말쯤 개발이 완료될 것으로 보인다.

드루팔 설치

<http://drupal.org/documentation/install>에서 드루팔 배포판을 받으면 그 안에 설치 방법이 수록된 INSTALL.txt 파일이 들어 있다.

드루팔 프로그래밍 샘플 코드

드루팔을 설치했다면 드루팔 프로그래밍 샘플 코드를 볼 수 있게끔 <http://drupal.org/project/examples>에서 Examples for Developers 프로젝트를 받아야 한다. 여기엔 드루팔 코어(애드온을 제외한 드루팔의 기본 시스템)와 더불어 다양한 드루팔 샘플 코드가 들어 있다.

드루팔 샘플 모듈

샘플 프로젝트 코드도 살펴봤으면 그 프로젝트의 샘플 모듈을 한두 개 설치해서 사용해 보고 코드를 훑어보자. 포함된 프로젝트 모듈의 설치법도 역시 README.txt 파일에 들어 있다. 유난히 끌리는 주제가 있다면 그 주제와 관련된 모듈을 택하면 된다. Block 샘플 모듈과 Page 샘플 모듈이 기본에 충실하므로 추천한다. 샘플 모듈 코드를 가지고 공부할 때, 특정 드루팔 API 함수의 설명을 보고 싶으면 드루팔 API 매뉴얼 공식 사이트인 <http://api.drupal.org>를 참고하자.

예제 파일

이 책에서 사용된 예제 파일은 <http://examples.oreilly.com/0636920027799/>에서 받을 수 있습니다.

한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1. eBook Only: 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다.

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수분에게는 보다 쉽게 집필하실 기회가 되리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2. 무료로 업데이트되는, 전자책 전용 서비스입니다.

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정한 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나, 저자(역자)와 독자가 소통하면서 보완되고 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3. 독자의 편의를 위하여, DRM-Free로 제공합니다.

구매한 전자책을 다양한 IT기기에서 자유롭게 활용하실 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해, 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4. 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다.

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 계실 가능성이 높습니다. 이 경우 저작권법에 의해 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한 한빛미디어 사이트에서 구입하신 후에는 횡수에 관계없이 다운받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 교정이나 내용의 수정보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구입하신 독자에게만 제공되며, 다른 사람에게 양도나 이전되지 않습니다.

차례

01	드루팔 소개	1
<hr/>		
	1.1 드루팔이 무엇인가.....	1
	1.1.1 코어.....	2
	1.1.2 애드온: 모듈, 테마, 배포판, 번역판.....	3
	1.2 URL 요청 처리 방식.....	4
	1.3 캐시.....	7
02	드루팔 코딩 원칙	9
<hr/>		
	2.1 원칙1_ 수정하기 쉬워야 한다.....	9
	2.1.1 모듈과 테마에 훅을 사용할 것.....	11
	2.1.2 완성한 코드는 테마를 적용할 수 있을 것.....	14
	2.2 원칙2_ 전 세계 언어를 지원해야 한다.....	17
	2.2.1 사용자 인터페이스 문구를 국제화할 것.....	18
	2.2.2 사용자 입력 문구를 국제화할 것.....	20
	2.3 원칙3_ 접근성과 활용성이 뛰어나야 한다.....	22
	2.4 원칙4_ 데이터베이스에 종속되지 말아야 한다.....	24
	2.4.1 데이터베이스 테이블은 Schema API와 hook_updateN()으로 구성할 것.....	26
	2.4.2 데이터베이스 질의엔 Database API를 이용할 것.....	29
	2.5 원칙5_ 드루팔 자체는 보안하되, 사용자 입력은 보안하지 말아야 한다.....	33
	2.5.1 사용자 입력 내용을 정리하고 확인할 것.....	34
	2.5.2 권한을 확인할 것.....	36
	2.6 원칙6_ 코드는 테스트를 거치고 문서화해야 한다.....	39

3.1 실수1_ 쓸데없이 길게 코딩한다.....	42
3.1.1 필드화 데이터를 사용해 자의적 코딩을 방지할 것.....	47
3.1.2 블록 배치를 위한 테마 영역을 정의할 것.....	50
3.2 실수2_ 드루팔 API를 잘못 사용한다.....	52
3.2.1 실수2-1_ 페이지가 로딩될 때마다 코드를 실행한다.....	52
3.2.2 실수2-2_ 범용 훅을 과다하게 이용한다.....	53
3.3 실수3_ 데이터베이스 작업에 PHP 코드를 아낀다.....	54
3.4 실수4_ 혼자서 작업한다.....	56
3.4.1 팀에 들어가거나 그룹으로 작업할 것.....	57
3.4.2 커뮤니티에 문제를 보고하고 코드를 기부할 것.....	58
3.4.3 다양한 방식으로 커뮤니티에 기여할 것.....	60

4.1 URL을 등록하고 콘텐츠를 표시하자.....	62
4.1.1 URL을 등록.....	63
4.1.2 등록된 URL을 변경.....	66
4.1.3 블록을 등록.....	67
4.1.4 페이지와 블록 출력물을 반환.....	69
4.1.5 Form API를 사용해 폼을 생성.....	74
4.2 엔티티와 필드를 코딩하자.....	85
4.2.1 엔티티와 필드.....	85
4.2.2 엔티티 타입을 정의.....	87

4.2.3	필드 타입을 정의.....	98
4.2.4	필드 위젯을 코딩.....	100
4.2.5	필드 포매터를 코딩.....	104
4.3	Views 모듈 애드온을 작성하자.....	106
4.3.1	Views 프로그래밍 용어와 출력 결과물 생성.....	108
4.3.2	Views에 사용할 모듈을 설정.....	109
4.3.3	새 Views 데이터 출처를 정의.....	110
4.3.4	기존 Views 데이터 출처에 필드와 관계를 추가.....	114
4.3.5	Views에 화면 출력 플러그인을 지정.....	116
4.3.6	기본 Views를 지정.....	119
4.4	Rules 모듈 애드온을 작성하자.....	121
4.4.1	Rules에 사용자 정의 기능을 지정.....	122
4.4.2	기본 Rules를 제공.....	125

05 코딩 도구와 참고사항 127

5.1	다양한 드루팔 개발 도구.....	127
5.2	드루팔 API 함수 찾기.....	129
5.3	그 밖의 코딩 팁과 조언.....	132

06 참고 자료 133

6.1	드루팔 사이트 제작과 일반적인 드루팔 관련.....	133
6.2	드루팔 프로그래밍 참고사항과 배경 지식.....	134
6.3	PHP 관련.....	136
6.4	데이터베이스 관련.....	136
6.5	그 밖의 웹 기술 관련.....	137

1 | 드루팔 소개

1.1 드루팔이 무엇인가

이 책을 든 여러분은 드루팔을 콘텐츠 관리 시스템(Content Management System, CMS 또는 콘텐츠 관리 프레임워크(Content Management Framework, CMF⁰¹)로 불리는 것을 듣게 될 것이다. CMS, CMF 둘 다 맞는 표현이다. 첫째, 드루팔을 CMS라고 할 수 있는 이유는 기본 드루팔 소프트웨어만 설치해도 포럼, 고정 페이지, 블로그로 구성된 웹사이트를 만들 수 있고 온라인에서 콘텐츠를 관리할 수 있기 때문이다. 둘째, 드루팔을 CMF라고 할 수 있는 이유는 사람들 대부분이 기능이 많은 복잡한 웹사이트를 제작하기 위해 부수적인 모듈을 선택해서 드루팔에 추가하며, 사용자도 자신만의 모듈을 만들 수 있기 때문이다.

드루팔은 무료로 제공되는 오픈 소스 소프트웨어(Free and Open Source Software, FOSS)이며, GNU 일반 공개 라이선스(GPL) 2 버전이나 자신의 선택에 따른 이후 버전이 적용된다. 드루팔을 사용할 예정인데 GPL을 읽어본 적이 없다면 어서 읽어보자. 자신이나 남이 사용할 드루팔 소프트웨어를 개발할 생각이라면 더욱 읽어야 한다. GPL은 드루팔 소프트웨어 자체를 가지고 해도 되는 일과 drupal.org에서 받은 애드온, drupal.org 문서 페이지에 있는 코드, 자신이나 남이 제작한 모든 파생 작업물(원본 그대로든 수정본이든 GPL 라이선스 작업물을 포함하든)을 가지고 해도 되는 일을 규제한다. GPL은 순수 영어로 작성되었고 프로그래머 입장에서 읽기도 좋게 돼 있다. GPL은 드루팔 코어와 함께 배포되는 LICENSE.txt 파일이나 <http://gnu.org> 사이트에서 읽을 수 있다.

끝으로, 드루팔은 프로젝트이자 커뮤니티다. 나중에 소스 코드를 대중에게 공개하는 한 업체에 의해 주로 개발이 이뤄지는 일부 무료 오픈 소스 소프트웨어와 달리, 드루팔 소프트웨어의 제작 및 테스트를 후원하고 문서를 작성하고 각국의 언어로 번역하고 질문에

01 역자주_자신만의 CMS를 제작하는 데 이용할 수 있는 발판

답변하고 drupal.org 웹 서버를 계속 가동하고 지역과 전 세계 규모로 모임을 준비할 수 있도록 시간과 돈을 후원하는 개인과 기업으로 구성된 국제 커뮤니티의 노력에 힘입어 드루팔은 지속적으로 발전해간다.

1.1.1 코어

드루팔 코어는 <http://drupal.org/project/drupal>에서 받는 드루팔 압축파일이며 PHP 스크립트(HTML 마크업이 들어 있는 것도 있음), 자바스크립트, CSS 등의 파일로 구성된다. 이 소프트웨어는 웹 서버(주로 아파치), 데이터베이스(MySQL, PostgreSQL, SQLite는 드루팔 코어 7 버전까지 지원되고, 다른 것들은 애드온 모듈을 통해 지원됨), 웹브라우저와의 정보 송수신을 통해 다음과 같은 CMS의 기본 기능을 제공한다.

- URL 요청 디스패치 시스템
- 유연한 권한과 역할이 존재하는 사용자 계정 관리 시스템
- 온라인 콘텐츠 편집
- 버튼부터 전체적인 페이지까지 모든 요소가 표시되는 방식을 재정의할 수 있는 테마(템플릿) 시스템
- 사이트 페이지의 각기 다른 영역에 있는 콘텐츠 덩어리들을 배치할 수 있는 블록 시스템(이 시스템은 드루팔 8 버전에서 상당히 달라지고 더 유연해질 예정임)
- 내비게이션 메뉴 빌더
- 카테고리, 태그, 분류 어휘의 사용자 정의 기능이 있는 유연한 분류 시스템
- 댓글, 콘텐츠 필드, RSS 수집, 검색, 포럼과 투표 같은 사이트 기능을 지원하는 각종 모듈. 이러한 모듈은 필수적이지 않으며 그 중에는 드루팔 코어에 포함되지 않아 애드온 모듈을 받아 설치해야 하는 것도 있다.
- 사이트를 다양한 언어로 구성하고 콘텐츠를 번역하는 기능(드루팔 버전에 따라 다르며, 다국어나 비영문 사이트를 제작할 때는 일부 애드온 모듈이 필요할 수도 있다)
- 시스템 이벤트와 예러의 기록
- 드루팔 프로그래머를 위한 API

1.1.2 애드온: 모듈, 테마, 배포판, 번역판

드루팔은 모듈화된 소프트웨어로 모듈을 활성화하거나 비활성화 하는 방식으로 사이트 기능을 켜고 끌 수 있다. 드루팔 코어에는 몇 가지 필수 모듈과 옵션 모듈이 포함돼 있다. 이외의 수천 가지 추가적인 기부 모듈은 <http://drupal.org/project/modules>에서 받을 수 있다. 대부분의 모듈은 설정 옵션이 있어서, 사용자는 적절한 권한이 부여된 계정으로 드루팔 사이트에 로그인한 다음 드루팔 관리 인터페이스에서 설정을 변경할 수 있다. 권한 시스템은 유연하다. 첫째로, '역할'이라는 것을 정의할 수 있는데, 각 역할에 따라 특정한 권한이 부여된다. 권한은 모듈에서 정의된다. 둘째로, 각 사용자 계정마다 하나 이상의 역할을 할당할 수 있다.

드루팔은 특정 HTML 마크업과 스타일 지정을 분리하는 테마 시스템을 이용한다. 따라서 사이트의 레이아웃이나 스타일을 변경하고 싶으면 <http://drupal.org/project/themes>에서 새 테마를 받거나, 상용 테마를 구입해 스스로 제작해서 사용하면 된다. 테마를 설치해서 활성화하면 콘텐츠 페이지를 수정하지 않아도 사이트의 외관이 즉시 변경된다. 테마 시스템을 이용하면 수정하려는 부분을 덮어쓰는 방식으로 기본 화면을 표시할 수 있다. 버튼의 표현물 같은 최저 수준의 요소에서부터 전체 페이지에 이르기까지 어느 요소든 덮어쓸 수 있다.

드루팔을 배포판으로 받을 수도 있다. 배포판은 드루팔 코어와 다수의 기부 모듈/테마로 구성된다. 이것들은 서로 작용하여 특정한 목적을 이행하는 더욱 기능적인 사이트를 형성한다. 배포판은 전자상거래, 정부, 비영리를 비롯한 다양한 목적으로 사용할 수 있으며 <http://drupal.org/project/distributions>에서 받을 수 있다.

끝으로, 드루팔과 기여 모듈/테마와 배포판의 번역본은 <http://localize.drupal.org>에서 받을 수 있다. 현재 드루팔 7 버전에서 번역본 설치하는 모듈이나 테마를 설치하는 것보다 더 복잡하며, 다국어 사이트를 만들 경우엔 더 복잡하다. 드루팔 8 버전에선 이 단점이 개선될 전망이다.

애드온 검색 방법

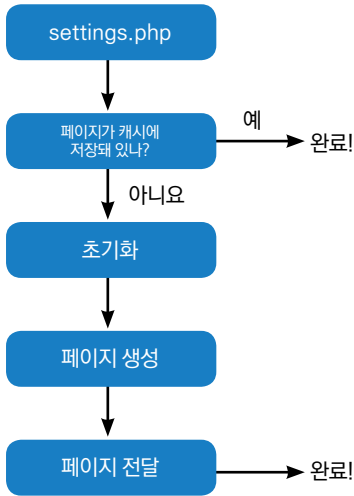
드루팔 애드온(모듈, 테마, 배포판)을 검색하는 주요 방법은 다음과 같다.

- 이름을 아는 특정 애드온을 검색하려면 <http://drupal.org>로 가서 검색 박스에 그 애드온 이름을 입력하자.
- 처음의 검색 결과에 그 애드온이 없다면, 우측 사이드 바에 있는 필터를 이용해 검색 조건을 모듈이나 테마로 한정하자. 현재는 검색 조건을 배포판으로 한정하는 기능이 없다.
- <http://drupal.org/project/modules> 또는 <http://drupal.org/project/themes> 또는 <http://drupal.org/project/distributions> 페이지로 가서 검색하자.
- 항상 drupal.org/project/ 뒤에 머신 네임(웹 주소 형식)이 붙으므로 URL을 추정해볼 수도 있다. 머신 네임은 소문자 알파벳, 숫자, 언더 바로 구성되지만 개발자가 정하므로 추측하기 어려워 두 배의 노력이 들 때도 있다. 예컨대, Views 모듈이 있는 페이지는 <http://drupal.org/project/views>이고, Pixture Reloaded 테마가 있는 페이지는 http://drupal.org/project/pixture_reloaded이고, XML Sitemap 모듈이 있는 페이지는 <http://drupal.org/project/xmlsitemap>이다.
- 찾을 애드온 이름을 모르면 <http://drupal.org/project/modules> 또는 <http://drupal.org/project/themes> 또는 <http://drupal.org/project/distributions>에서 키워드, 드루팔 버전 호환성, 카테고리(모듈만 가능) 등의 조건으로 검색하자.

1.2 URL 요청 처리 방식

드루팔이 제대로 설치된 상태에서 웹 서버가 드루팔 사이트에 알맞은 HTTP 요청을 받으면, 드루팔의 메인 `index.php` 파일이 로딩되고 그 요청을 처리할 서버에 의해 실행된다. 드루팔 프로그래머는 드루팔이 그런 요청을 어떻게 처리하는지 원리를 반드시 이해해야 한다. 그 원리를 요약하면 다음과 같다. [그림 1-1]도 참고하자.

그림 1-1 드루팔 HTTP 요청 처리 과정



1. 드루팔이 HTTP 요청에 어느 settings.php 파일을 사용할지를 결정하면, 그 파일이 로딩되어 실행된다. 각 사이트별 settings.php 파일을 사용하게 하면 드루팔이 멀티 사이트를 구현하게끔 설정할 수 있다.
2. URL 요청이 익명의 사용자(로그인하지 않은 사이트 방문자)로부터 오면, 드루팔은 페이지 캐시를 검사해서 그 요청된 URL에 대해 표시할 출력 결과가 미리 캐시에 저장돼 있는지 파악한다. 그래서 만약 저장돼 있다면 그 캐시에 저장된 출력 결과가 웹 서버로 반환되고, 드루팔은 종료된다. 드루팔 페이지 캐시 저장 기능은 인증받거나 로그인한 사용자에게만 적용되지 않는다.
3. 데이터베이스 접속, 설정/변수 시스템, PHP 세션 변수가 초기화된다.
4. 언어 시스템이 초기화된 후 각종 파일이 로딩되어 실행된다. 코어 인클루드 파일과 활성화된 모듈에 해당하는 module 확장자 파일들이다.
5. 드루팔은 사이트가 오프라인인지 온라인인지를 판단한다. 오프라인 상태를 유지 모드(maintenance mode)라고도 한다.
6. 사이트가 오프라인이면, 드루팔은 관리자가 저장한 오프라인 메시지를 가져와 페이지 콘텐츠로 표시한다. 그리고 그 밖의 함수들이 호출되어 페이지 콘텐츠의 일부분을 생성한다.

7. 사이트가 온라인 상태이거나 오프라인 중에 로그인한 사용자가 페이지에 접근하면, 드루팔은 그 요청에 대한 콘텐츠를 생성하려면 어느 함수들을 호출해야 할지 판단해서 호출한다. 이 함수들은 원칙적으로 미리 렌더링돼 있던 형식화하지 않은 콘텐츠를 반환하지만, 직접 전체나 일부분이 렌더링된 콘텐츠를 반환하는 것도 가능하다.
8. 드루팔은 페이지에 무슨 전달 메서드를 사용할지 판단해서, 적절한 전달 함수를 호출한다.
9. HTML 페이지를 요청받으면 기본 페이지 전달 함수는 HTTP 헤더를 print문으로 출력하고, 테마를 사용해서 HTML 안에 형식화하지 않은 콘텐츠를 렌더링한 후, HTML 결과물을 print문으로 출력하고(웹 서버로 전달하는 효과적인 방법), 사용자 세션 정보를 데이터베이스에 저장한 후, 종료된다. AJAX 요청 전달 함수도 이와 비슷하지만, 테마 시스템을 이용해 HTML을 렌더링하는 게 아니라 JSON 결과물로 렌더링한다는 점이 다르다. 모듈은 사용자 정의 페이지 전달 메서드도 정의할 수도 있다.

이와 관련된 주제를 설명한 절은 다음과 같다.

- 1장의 '캐시' 절(7페이지)
- 4장의 '페이지와 블록 출력물을 반환' 절
- 6장의 '그 밖의 웹 기술 관련' 절: 웹 기술 부분의 참고 자료로 웹 서버가 일반적으로 요청을 처리하는 방식을 공부할 수 있다.

드루팔 8

지금까지 설명한 추상적인 내용의 요약 설명 중 일부는 드루팔 8 버전에도 적용되지만, 각 단계의 세부적인 사항은 많이 달라질 수도 있다. 특히 로딩되는 파일 수가 줄어들 것이며, 철학적인 관점에서 드루팔은 주어진 URL에 대한 HTML 페이지를 반환하는 게 아니라 세션 변수와 다른 문맥 정보를 포함한 범용 HTTP 요청에 대한 응답을 지향하는 방향으로 발전할 것이다.

1.3 캐시

드루팔에는 캐시 시스템이 들어 있다. 캐시 시스템 덕분에 모듈은 데이터나 출력할 결과물을 미리 계산해서 그 결과를 다음번에 필요할 때 다시 계산하지 않아도 되게끔 데이터베이스에 저장해둔다. 이로써 복잡도는 증가하지만 페이지 로딩 시간이 크게 줄어든다. 캐시 기능을 이용하는 모듈은 A 데이터가 의존적인 B 데이터의 변화로 인해 사용할 수 없을 때마다 캐시에 저장된 데이터를 반드시 제거해야 한다. 드루팔 7의 캐시 시스템은 아주 단순한 API 하나로 이뤄지는데, 그 API는 `cache_set()` 함수와 `cache_get()` 함수를 비롯한 몇 가지 변종 함수들과 데이터베이스 캐시 및 모듈 전용 캐시를 포함해 전부 초기화하는 `cache_clear_all()` 함수로 구성된다. 모듈은 `hook_flush_caches()` 라는 훅을 구현해서 자체 캐시가 초기화되게끔 등록할 수 있다. 훅은 드루팔을 수정하기 위한 모듈의 진입점이다.

드루팔 코어와 애드온 모듈은 이러한 캐시 시스템을 이용해 정보를 캐시에 저장한다. 이에 대한 몇 가지 예는 다음과 같다.

- 익명 사용자에게 표시할 페이지 결과물(페이지 캐시 기능은 Performance 설정 페이지에서 끌 수 있다)
- 블록 결과물(블록 캐시 기능은 Performance 설정 페이지에서 끌 수 있다)
- 모듈로부터 수집된 메뉴 라우팅(URL) 정보, 블록 등록 정보, 그 밖의 정보
- 모듈과 테마에 있는 테마 영역들과 테마 관련 정보를 포함한 테마 정보
- 폼 배열

드루팔에 입문하는 프로그래머와 사이트 제작자는 사이트에 문제가 생기거나 최근에 수정한 코드를 알아볼 수 없을 때 가장 먼저 해 봐야 할 일이 캐시를 초기화하는 일임을 금방 알게 된다. 캐시를 초기화하려면 Performance 설정 페이지로 가서 캐시 초기화 버튼을 클릭하든지 Drush를 이용하면 된다.

이와 관련된 주제, 예제, 참고 자료가 있는 절은 다음과 같다.

- 테마 혹은 비릇한 훅에 관한 일반적인 설명은 2장의 '원칙1_수정하기 쉬워야 한다' 절을 참고하자.
- 이 장에 언급한 캐시 함수들에 대한 완전한 설명은 <http://api.drupal.org>를 참고하자.
- Drush에 대해 자세히 알고 싶다면 5장의 '다양한 드루팔 개발 도구' 절을 참고하자.
- Cache API 사용법을 알아보려면 Examples for Developers 프로젝트 페이지에 있는 예제 (<http://drupal.org/project/examples>) 중 Cache 예제를 참고하자.
- 메뉴 라우팅 캐시에 관해서는 4장의 'URL을 등록' 절과 Examples for Developers 페이지에 있는 Page 예제를 참고하자.
- 블록 등록 캐시에 관해서는 4장의 '블록을 등록' 절과 Examples for Developers 페이지에 있는 Block 예제를 참고하자.
- 뷰 캐시에 관해서는 4장의 'Views 모듈 애드온을 작성하자' 절을 참고하자.
- 테마 캐시에 관해서는 3장의 '블록 배치를 위한 테마 영역을 정의할 것' 절과 2장의 '완성한 코드는 테마를 적용할 수 있을 것' 절을 참고하자.
- 폼에 관해서는 4장의 'Form API를 사용해 폼을 생성' 절과 Examples for Developers 페이지에 있는 Form 예제를 참고하자.

드루팔 8

Cache API 함수들은 드루팔 8 버전에서 클래스를 이용해 캐시를 관리할 수 있도록 바뀌었다.

2 | 드루팔 코딩 원칙

경력 프로그래머는 연습과 실험을 통해 각종 원칙과 최선의 사례를 배우고, 개발하면서 해결해야 할 문제에 부딪혔을 때 그렇게 배운 내용을 적용한다. 여기엔 ‘코드에 주석을 달아라’ 또는 ‘변수명은 간결하게 정해라’ 같이 프로그래밍 언어와 상황에 관계없이 항상 적용되는 일반적인 관례와 특정 영역에 국한적으로 적용되는 일부 관례도 포함된다. 드루팔은 몇 가지 고유한 프로그래밍 원칙이 존재하는데, 이 장에서 그 원칙들에 대해 알아볼 것이다. 이 원칙들을 습득하고 준수한다면 여러분은 틀림없이 더 효율성 높은 드루팔 프로그래머로 변신할 것이다.

드루팔 프로그래밍이 처음인 독자는 이 장을 읽기 전에 <http://drupal.org/project/examples> 에서 Examples for Developers 프로젝트를 받아서 살펴보면 도움이 될 것이다. Page 예제와 Block 예제 코드를 직접 실행하고 소스 코드를 살펴봄으로써 맥락을 좀 더 파악한 후에 이 장을 읽으면 내용을 이해하기 쉬울 것이다.

2.1 원칙1 _수정하기 쉬워야 한다

드루팔의 용도가 애초에 웹 애플리케이션 제작용 플랫폼이란 점에서, 기본 원칙 하나는 드루팔에 관련된 거의 모든 것은 사용자가 원하는 대로 수정하기 쉬워야 한다는 것이다. 단, 기반이 되는 소스 코드에 손대지 않고 원하는 대로 수정할 수 있어야 한다. 드루팔로 어떤 종류의 웹 애플리케이션을 제작하든 간에 기반 코드를 수정하지 않고 드루팔 코어와 기부 모듈을 완전히 수정 가능해야 하므로, 이를 위해 코어와 기부 모듈에는 기능과 외형을 사용자가 정의할 수 있는 훅(hook)이 존재한다.

‘훅’이라는 용어는 여러 CMS 프로젝트에서 비슷한 의미로 쓰인다. 훅이란, 애드온 모듈이나 플러그인이 기반 CMS의 기능이나 외형을 변경할 수 있게 작용하는 진입점을 뜻한다. 드루팔에서는 특히 다소 중복되는 종류의 훅이 몇 개 존재한다.

- **범용 훅:** 모듈은 이 훅을 통해 드루팔 기능에 다른 기능을 추가로 정의할 수 있다.
- **수정 훅:** 모듈은 이 훅을 통해 드루팔의 기존 기능에 수정을 가할 수 있다.
- **테마 훅:** 테마는 이 훅을 통해 브라우저로 전송된 결과물을 수정할 수 있다.
- **전처리 훅과 처리 훅에 딸린 테마 훅:** 모듈은 이 훅을 통해 출력을 위해 테마로 전송된 정보를 수정할 수 있다.

그냥 코드를 해킹(수정)하면 안 될까?

다른 프로그래밍 커뮤니티에서와 다르게, 드루팔 커뮤니티에서는 ‘해킹^{hack}’이라는 용어가 확실히 부정적인 의미로 쓰인다. 드루팔에서 해킹은 구체적으로 사이트를 필요에 맞게 고치려고 Drupal.org에서 받은 코드를 수정하는 것을 뜻한다. 해킹은 아주 바람직하지 않다. ‘코어를 해킹하는 일’, 즉 드루팔 코어 파일을 수정하는 일은 최후의 보루로 고려해야 한다. 그해야 하는 몇 가지 이유는 다음과 같다.

- 해킹은 대체로 불필요하다. 해킹하지 않고도 목적을 달성할 수 있는 훅이 분명 존재할 것이기 때문이다.
- 드루팔 코어나 드루팔 홈페이지에서 받은 다른 코드를 해킹하면 나중에 업데이트 작업이 훨씬 복잡해진다. 새 버전의 코드를 받을 때마다 자신이 적용했던 핵을 매번 다시 적용해야 하기 때문이다. 드루팔 홈페이지에서 받은 코드는 전부 보안 수정, 버그 수정, 새 기능 추가 등의 이유로 적어도 가끔은 업데이트된다.
- 해킹 대신 훅을 사용하면, 수정 사항을 적용하고 싶지 않을 땐 그 훅이 들어 있는 모듈만 비활성화하면 된다. 반면에, 코드를 해킹하면 수정 사항을 적용하고 싶지 않을 땐 해킹한 부분을 원래의 코드로 되돌려야 해서 복잡하고 힘들다.
- 자신에게 필요한 훅이 존재하지 않든지, 모듈에 버그 수정이나 새 기능 추가를 위해 핵을 사용하고 싶을 땐, 핵을 패치로 전환하면 그 패치가 모듈이나 드루팔 코어에 추가될 경우 자신과 드루팔 커뮤니티에 이익이 된다.

그 밖의 참고 자료는 다음과 같다.

- 드루팔 코어의 범용 훅과 수정 훅을 전부 보려면 <http://api.drupal.org>의 Hooks 부분을 참고하고, 드루팔 코어 테마 훅을 전부 보려면 ‘Default theme implementations’ 부분을 참고하자.
- 패치에 대해 알고 싶다면 3장의 ‘커뮤니티에 문제를 보고하고 코드를 기부할 것’을 참고하자.

드루팔 8

드루팔 8 버전도 분명히 수정은 용이하겠지만, 8 버전은 혹은 새 플러그인 시스템을 병용할 것으로 전망된다.

2.1.1 모듈과 테마에 혹은 사용할 것

기본적으로(적어도 드루팔 프로그래머 관점에서와 드루팔 7에서 업데이트한 버전에서는) 모듈은 혹은 구현 코드(혹의 외형이나 기능을 정의하는 PHP 함수)를 위주로 모아놓은 것이며, 테마는 테마 혹은 재정의의 코드(결과물의 표현 방식을 정의하는 PHP 함수나 템플릿 파일)를 위주로 모아놓은 것이다. 테마와 모듈에는 보조 코드, CSS, 자바스크립트, 이미지를 비롯한 각종 파일도 들어 있을 수 있다. 드루팔을 고치는 목적이 HTML과 CSS 같은 데이터의 외적인 표현을 변경하는 것이라면, 보통 테마에 들어 있는 테마 혹은 재정의해야 한다. 그리고 자신의 목적이 드루팔의 동작 방식이나 출력될 데이터를 변경하는 것이라면, 모듈 안에 혹은 구현해야 한다. 외형과 기능을 둘 다 변경해야 할 때는 테마와 모듈의 혹은 둘 다 손봐야 한다.

모듈이나 테마를 설정하고 혹은 구현하거나 재정의하는 절차는 다음과 같다.

1. 머신 네임이나 모듈/테마의 단축명을 선택하자. 이런 이름은 보통 알파벳, 언더 바, 숫자가 이어진 형태이며 함수 접두어로 사용될 것이기 때문에 PHP의 함수명 정하기 규칙에 부합해야 한다(머신 네임을 모듈이나 테마 안에 자신이 정의할 모든 함수의 접두어로 사용하자). drupal.org 홈페이지의 프로젝트에 사용되고 있지 않은 이름으로 정해야 나중의 혼동을 피할 수 있다. 이 명명 규칙에 따라 이 책에서는 mymodule이라는 이름의 모듈이나 mytheme라는 이름의 테마를 작성한다.
2. 테마나 모듈이 들어갈 디렉터리를 만들자. 이 디렉터리를 어느 위치에 만들지 모르겠다면 뒤에 나올 '모듈과 테마를 어느 디렉터리에 넣지?' 박스를 참고하자.
3. 모듈 디렉터리 안에는 mymodule.info 파일을 테마 디렉터리 안에는 mytheme.info 파일을 만들자. 이 파일은 모듈과 테마에 관한 정보가 담긴 일반 텍스트 파일이다. 이 파일의 정확한 문법은 드루팔 버전마다 다르며 모듈과 테마에 따라서도 약간씩 다르다. 따라서 온라인 문서를 참고하든지, 처음

엔 드루팔 코어나 Examples for Developers 프로젝트에 든 모듈이나 테마의 파일을 복사해서 쓰자. 드루팔 7 모듈의 간단한 예는 다음과 같다. 테마 파일의 문법도 이것과 대부분 같다.

```
; 주석 맨 앞에는 이렇게 세미콜론을 붙인다.  
; Modules나 Themes 목록에 표시될 이름  
name = My Module  
; Modules나 Themes 목록에 표시될 상세 설명  
description = 이것은 이 모듈에 대한 자세한 설명이다.  
; 이 모듈이나 테마와 호환되는 드루팔 코어 버전  
core = 7.x
```

4. 테마 혹은 함수나 모듈 혹은 함수를 작성하기 위해, 테마 디렉터리나 모듈 디렉터리 안에 혹은 구현이나 재정의의 함수를 작성할 PHP 파일을 생성하자(테마 템플릿은 이 단계를 생략). 테마 혹은 함수가 들어갈 파일의 이름은 template.php이다. 대부분의 범용 모듈 혹은 함수가 들어갈 파일의 이름은 mymodule.module이지만 일부 혹은 구현 함수는 다른 파일에 들어 있다. 그 예로, hook_schema() 함수와 hook_update_N() 함수 같이 설치와 관련된 혹은 구현 함수는 mymodule.install 파일에 들어 있고, 기부 Views 모듈은 혹은 서로 다른 두 파일에 들어 있다. 자신이 구현할 혹은 대해 설명된 문서를 항상 보면서 구현 함수가 어느 파일에 들어 있는지 파악하자.
5. 모듈 파일이나 테마 파일 안에, 혹은 구현하거나 테마 함수를 재정의하는 함수를 정의하자(테마 템플릿은 이 단계 생략). hook_foo() 혹은 함수의 구현/재정의의 함수명은 반드시 mymodule_foo() 라는 이름이어야 하고, theme_foo() 테마 혹은 함수의 구현/재정의의 함수명은 반드시 mytheme_foo()라는 이름이어야 한다. 함수 안에 작성할 내용은 혹은 문서를 먼저 읽거나 재정의하려는 테마 혹은 함수를 참고하면 대부분 알 수 있다.
6. 테마 혹은 템플릿 파일을 재정의 하기 위해, 재정의할 템플릿 파일을 테마 디렉터리로 복사하자. 이때 파일명은 바뀌지 않아야 한다.
7. 함수나 파일을 수정해서 원하는 대로 변경하자.
8. 모듈이나 테마를 활성화하자.
9. 코딩과 테스트 중에, 혹은 활성화돼 있는 상태의 모듈이나 테마에 새 혹은 구현 함수나 테마 재정의의 함수를 추가했으면, 드루팔 캐시를 초기화해야 그 변경 사항이 반영된다. 일부 정보 제공용 혹은 경

우엔 함수 내용을 수정한 후에도 캐시를 초기화해야 드루팔이 수정한 훅을 재실행해서 새로 바뀐 정보를 읽어 들인다. 일반적으로 개발 중에는 무엇을 수정하던 캐시를 초기화하는 습관을 들이는 것이 좋다.

훅 구현 함수의 명명 규칙을 바꿔서 드루팔 8이나 다른 차후의 버전부터는 `mymodule_foo()` 같이 단어를 언더 바 하나가 아니라 두 개로 연결하자는 일각의 논의가 있었다. 왜냐하면 많은 모듈과 훅의 이름에 언더 바가 들어 있으므로, 함수명엔 단어 구분자로 이중 언더 바를 써야 함수와 모듈/훅을 더 쉽게 구별할 수 있지 않겠냐는 이유 때문이다. 그러나 일단 현재는 아직 이 새 규칙은 채택되지 않았다.

그 밖의 참고 자료와 예제 코드는 다음과 같다.

- 모듈용 info 확장자 파일의 문법이 궁금하면 <http://bit.ly/U0SuhH>를 참고하자.
- 테마용 info 확장자 파일의 문법이 궁금하면 <http://bit.ly/Tol9P7>을 참고하자.
- 1장의 '캐시'를 참고하자.
- 다양한 훅 문서는 <http://api.drupal.org>에 가서 `hook_block_info`, `theme_table`, `block.tpl.php` 같은 훅 함수명이나 템플릿 파일명으로 검색하면 찾을 수 있다. 훅 문서 페이지에는 훅의 목적, 매개변수, 반환 값, 훅 함수를 어느 파일에 넣어야 할지, 훅이 호출되는 위치 목록(개발자의 훅 구현 함수를 드루팔 시스템이 호출함), 드루팔 코어에 포함된 다양한 구현 함수들이 잘 설명돼 있다. 구현 함수를 작성할 때는 훅 함수의 내용을 샘플로 참고하면 된다. 드루팔 홈페이지의 테마 훅 페이지에 가면 테마 훅이 어디에 사용되는지 잘 설명돼 있고, 결과물에 사용할 수 있는 변수들에 대해서도 설명돼 있다.
- 데이터베이스 스키마와 설치에 관련된 각종 훅이 궁금하면 2장의 '데이터베이스 테이블은 Schema API와 `hook_update_N()`으로 구성할 것'을 참고하자.
- Views 모듈의 각종 훅이 궁금하면 4장의 'Views 모듈 애드온을 작성하자'를 참고하자.
- 자신의 모듈이나 테마에 사용 예정인 머신 네임이 이미 사용 중인 이름인지 알아보려면, 브라우저 주소창에 `'drupal.org/project/자신의_사용_예정_이름'`을 입력해서 확인하자.

모듈과 테마를 어느 디렉터리에 넣지?

드루팔 7 이하 버전에선, 드루팔 홈페이지에서 받거나 직접 작성한 모듈과 테마를 보통은 sites/all/modules 디렉터리와 sites/all/themes 디렉터리에 넣어야 한다. 각 모듈/테마 프로젝트는 모듈/테마 디렉터리의 하위 디렉터리에 넣어야 한다(예: Views 프로젝트라면 sites/all/modules/views). 하위 디렉터리 안에 각종 모듈을 정리해 넣을 수도 있다. 예를 들어 sites/all/modules/contrib 디렉터리를 만들어 사이트에서 받거나 남이 만들어 기부한 모듈만 모아놓고 sites/all/modules/custom 디렉터리를 만들어 자신이 수정했거나 만든 모듈만 모아놓을 수도 있다.

멀티 사이트를 설치했고 모듈이나 테마를 그 멀티 사이트 중 한 곳에서만 사용할 수 있게 하려면, 모듈은 sites/특정한_사이트/modules 디렉터리에 넣고 테마는 sites/특정한_사이트/themes 디렉터리에 넣으면 된다. 여기에 관한 자세한 설명은 <http://drupal.org/documentation/install/multi-site>를 참고하자.

드루팔 8 버전에선 인클루드 파일, 모듈, 테마를 비롯한 드루팔 코어 코드 전체가 core 디렉터리로 옮겨졌다. 그리고 받은 모듈과 자신이 제작/수정한 모듈을 기존 그대로 sites/all 디렉터리에 넣어도 되지만, 8 버전에선 최상위 모듈/테마 디렉터리도 새로 생겼다. 멀티 사이트 설치 시의 특정 사이트 전용 모듈/테마를 제외하고는 이렇게 새로 생긴 최상위 디렉터리를 사용하는 것이 더 좋다.

2.1.2 완성한 코드는 테마를 적용할 수 있을 것

드루팔의 테마 시스템은 데이터/콘텐츠와 스타일/표현이 분리되게끔 설계됐다. 모듈은 데이터/콘텐츠를 제어하는 기능을 갖춰야 하고, 사용 중인 테마는 데이터를 HTML로 렌더링하는 것을 포함해 스타일/표현을 전반적으로 제어하는 기능을 갖춰야 한다. 이것이 가능하려면, HTML로 렌더링될 모든 데이터가 반드시 드루팔 theme() 함수를 통해 전달돼야 한다는 기본 원칙을 지켜야 한다. 예를 들면 코어 Block 모듈을 구성하는 한 블록이 렌더링될 때마다, Block 모듈이 단순히 데이터를 감싼 HTML의 <div>를 출력하는 게 아니라 theme('block', \$block_data) 함수가 호출된다. 이렇게 하면 자신이 작성한 테마로 기본 테마 템플릿 파일인 block.tpl.php를 덮어쓸 수 있다. theme() 함수가 받는 첫 번째 인자는 테마 혹은 함수명이고, 두 번째 인자는 테마 혹은 함수가 렌더링을 수행하는 데 필요한 데이터와 속성들로 구성된 배열이다.

테마 함수 Theme function와 렌더 배열 render arrays

많은 이들이 테마 함수와 렌더 배열의 관계를 혼동한다. 렌더 배열은 출력할 데이터가 담긴 배열이다. 기본 발상은 렌더 배열의 각 원소가 테마 함수나 테마 템플릿에 의해 렌더링(HTML 문자열로 변환)되며, 렌더링은 페이지 생성 처리 중에 최대한 늦게 이뤄져야 한다는 것이다. 따라서 개발자는 일반적으로 블록과 페이지 결과물을 제공하는 함수들에서 렌더 배열을 반환해야 하며, 필요하다면 결과 렌더 배열의 원소들을 렌더링하는 테마 함수나 템플릿을 정의해야 한다.

개발자는 모듈을 바로 그 원칙에 따라 작성해야 한다. 즉 모듈이 데이터를 HTML로 출력할 경우 그 모듈은 HTML을 직접 생성하는 게 아니라 드루팔의 테마 시스템을 이용해 결과를 렌더링해야 한다. 이 원칙을 단계별로 요약하면 다음과 같다. Examples for Developers 프로젝트에서 Theming 예제를 보면서 다음 과정을 따라 해보자.

1. 생성하려는 종류의 결과를 얻기 위한 테마 훅이 이미 존재하는지 확인하자. 예를 들어 만약 HTML 테이블을 생성하는 것이 목적이라면 자신이 직접 훅을 작성하지 말고 이미 있는 드루팔 코어의 table 테마 훅을 사용해야 한다.
2. 목적하는 바를 성취할 수 있는 테마 훅이 이미 존재하지 않을 땐, 다음과 같이 mymodule.module 파일에 있는 hook_theme() 함수를 구현해서 자신의 테마 훅을 정의하자.

```
function mymodule_theme($existing, $type, $theme, $path) {
  return array(
    // 배열의 키는 정의하는 테마 훅의 이름과 같다
    'mymodule_hookname' => array(
      // 입력 변수들
      'variables' => array(
        // 이 변수들은 theme() 함수에 배열로 전달되고, theme() 함수는
        // 그 변수들을 다시 개발자의 테마 함수나 템플릿에 전달한다.
        // 여기엔 기본 값을 할당한다.
        'input1' => '',
      ),
    ),
  );
}
```