|  |  |
| --- | --- |
| **알고 쓰자, 바르게 쓰자, 오픈소스 강의 스크립트** |  |

본 강의 스크립트는 사내 직원들을 위해 오픈소스 라이선스 강의를 해야 하는 사내 강사용으로 작성하였습니다.

강의 스크립트는 하나의 예시일 뿐, 여러분들의 회사 상황에 맞도록 커스터마이징해서 강의 스크립트를 만들어 보세요.

**Chapter 1: 저작권과 오픈소스 라이선스 (15분)**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **탭** | **강의 스크립트** | **비고** |
| 0 | **강사소개** | 안녕하세요. 저는 OOO에서 근무하고 있는 OOO입니다.  오늘 여러분들께 “알고 쓰자, 바르게 쓰자 오픈소스” 라는 주제로 교육을 하게 되었습니다.  대부분은 들어보신 내용일텐데요. 이번 시간을 통해서 확실하게 알게 되셨으면 좋겠습니다. | 상단에는 회사 로고를 삽입합니다. |
| 1 | **학습목표와 학습내용** | 이 교육을 모두 수강하게 되면 여러분들은  오픈소스의 정의와 주요 오픈소스 라이선스를 설명할 수 있게 될 거구요.  회사에서 사용을 허용하는 라이선스와 사용을 금지하는 라이선스를 구분할 수 있게 됩니다. 마지막으로 코드 작성 시 준수해야 할 사항도 알 수 있게 됩니다. |  |
| 2 |  | 본 교육에서는 오픈소스 코드를 사내 소스코드에 삽입하는 것에 대한 것만 다룹니다.  **[클릭]** 오픈소스 바이너리를 다운받아 PC에 설치하는 것에 대한 것은 다루지 않습니다. |  |
| 3 | **Chapter 1 제목**  **저작권과 오픈소스 라이선스** | 첫 번째 챕터인 “저작권과 오픈소스 라이선스” 에 대해 알아보겠습니다. |  |
| 4 | **오픈소스의 개념** | **자, 그럼 오픈소스의 개념부터 짚고 넘어가야겠죠?**  오픈소스란 저작권자가 누구나 자유롭게 사용 및 수정, 재배포할 수 있도록 공개한 소프트웨어입니다.  오픈소스에 대해 많은 분들이 가지고 계신 인식이 오픈소스는 “무료” 라는 건데요, Free & Open 소스에서 Free & Open은 “무료”를 의미하는 것이 아니라 소스코드의 자유로운 사용을 의미합니다. |  |
| 5 |  | 오픈소스와 상용 소프트웨어를 비교해 보겠습니다.  오픈소스는 소스코드가 제공되므로 코드의 사용, 복제, 배포, 수정이 가능합니다. 제가 앞에서 오픈소스가 왜 무료가 아니라고 했냐면 기술지원을 받기 위해서 비용을 지불하는 경우도 있기 때문입니다.  상용 소프트웨어는 대부분은 실행파일만 제공되기 때문에 직접 수정이나 배포를 할 수 없습니다.  여기서 프리웨어라는 개념이 나오는데요. 프리웨어는 상용 소프트웨어 중에서 무료로 사용할 수 있는 소프트웨어를 의미합니다. 상용 소프트웨어 범주 안에 프리웨어가 속하는거죠. “곰오디오“와 같은 프로그램이 대표적인 프리웨어라고 할 수 있습니다. |  |
| 6 |  | 오픈소스와 상용 소프트웨어의 예를 들어 보겠습니다.  OS 중에서 대표적인 오픈소스는 리눅스가 있고, 상용 소프트웨어에는 윈도우가 있습니다. |  |
| 7 |  | 데이터베이스에서 오픈소스는 MariaDB가 있고, 상용 소프트웨어에는 MSSQL 서버가 있습니다. |  |
| 8 |  | 이렇게 오픈소스와 상용 소프트웨어가 차이점들이 있다고 해도, 둘 다 지적재산권법으로 보호를 받고 있고 라이선스를 위반하게 되면 손해배상 또는 형사처벌을 받을 수 있다는 점에서는 공통점도 있습니다. |  |
| 9 | **오픈소스 라이선스의**  **주요 의무사항** | **그러면 오픈소스 라이선스의 주요 의무사항에 대해 살펴보겠습니다.**  제가 앞에서 라이선스 라는 말을 계속 하고 있는데, 오픈소스 라이선스는 저작권자의 독점 사용 권리 대해 저작권자와 사용자간의 이용방법 및 조건 범위를 명시해 놓은 사용 허가권입니다.  이러한 라이선스 전문에는 이 오픈소스로 무엇을 할 수 있는지, 사용하는 사람은 무엇을 준수해야 해야 하는지 등이 다 적혀 있습니다. |  |
| 10 |  | 라이선스는 저작권자가 직접 정할 수도 있지만, 보통 기존에 만들어져서 많이 사용하고 있는 것 중에서 선택하는 경우가 일반적입니다.  한 번쯤 들어본 적이 있을텐데요. MIT 라이선스, Apache 라이선스, GPL 라이선스라고 하는 것들이 다 이런 오픈소스 라이선스들입니다. |  |
| 11 |  | 라이선스에 따라 의무사항들은 조금씩 차이가 있는데, 일반적인 의무사항 몇 가지만 살펴보겠습니다.  가장 기본이 되는 것은 논문이나 책을 쓰고 나면 인용했던 참고문헌을 밝히는 것처럼 오픈소스를 사용하면 해당 오픈소스의 출처와 원저작자를 밝히는 것입니다. |  |
| 12 |  | 수정해서 사용하게 되면 어느 부분을 수정했다고 수정사실을 고지하기도 하고요. |  |
| 13 |  | 오픈소스 라이선스의 사본을 유지하고 첨부해야 합니다. |  |
| 14 |  |  | 15p의 샘플 페이지입니다. |
| 15 |  | 이러한 의무사항은 오픈소스를 이용해서 만든 파생저작물을 회사 외부로 배포할 때 발생합니다.  배포는 소스코드나 바이너리를 다른 사람 또는 다른 회사로 제공하는 것을 의미합니다. 회사에서 앱을 만들어서 고객들에게 배포하는 것이 가장 대표적인 배포의 예라고 할 수 있습니다. | 회사의 배포 사례를 직접 기입해서 설명하세요. |
| 16 |  | 그래서 회사 내부 사용 목적만으로 구축된 서비스나 앱을 개발할 때는 이런 의무사항이 적용되지 않습니다. 앱이 회사 외부로 배포되는 것이 아니니까요.  서버도 마찬가지입니다. 서버는 IDC나 AWS 같은 곳에 구축해 놓고 서비스를 하니까 배포가 아닙니다. |  |
| 17 |  | 근데, 해외 서비스를 위해서 퍼블리셔에게 서버 바이너리를 전달하게 되면 이게 바로 배포입니다.  인트라넷 서비스를 해외 지사에서 사용하고 싶어해서 지사에도 구축을 해주게 되면 이것도 배포가 되는 겁니다. |  |
| 18 |  | 다시 라이선스로 돌아와서, 아무 제한 없이 사용할 수 있는 public domain 이라는 것이 있습니다.  이건 저작권자가 저작권을 포기했거나, 저작권 보호기간이 지난 것을 의미하는데요. 아무 제약없이 마음대로 사용할 수 있습니다.  단, 저작권자가 누군지 알 수 없다고 해서 모두 public domain 이라고 단정짓고 무단으로 사용해서는 안 됩니다. |  |
| 19 | **위반 사례** | **다음은 오픈소스 라이선스를 위반하면 어떤 일이 발생할 수 있는지 사례를 보겠습니다.**  예전에 플레이스테이션2에서 있었던 사례인데요. ICO(이코) 라는 게임이 있었습니다. 어떤 이상한 사람이 (이름이 진짜로 알렉산더 스트레인저에요) 이코에 사용된 텍스터를 역분석했다가 GPL 라이브러리가 사용된 것을 발견했습니다. 그러면 개발사는 해당 소프트웨어의 소스코드를 공개하거나, 그게 싫으면 해당 소스를 수정을 해야 되는데 이미 게임은 많이 팔려나간 뒤였죠. 그래서 소스코드는 소스코드대로 공개하고 판매는 중단하는 최악의 결과를 낳고 말았습니다. |  |
| 20 | **주요 오픈소스 라이선스** | **다음은 주요 오픈소스 라이선스에 대해 살펴보겠습니다.**  오픈소스 라이선스는 전 세계적으로 약 2,500개 정도나 된다고 합니다. 지금도 만들어지고 있을거에요.  여기 보이는 것처럼 “네가 하고 싶은대로 해라” 라는 라이선스도 있어요.  그러면 2,500개를 다 알고 있어야 되느냐…! 그건 아니고요. |  |
| 21 |  | 사용 빈도 순으로 봤을 때 상위 10개의 라이선스가 전체 라이선스의 93%를 차지하고 있기 때문에 이 10개만 확실하게 알아두시면 됩니다.  맨 아래에 있는 AGPL 라이선스는 웹 서비스를 통해 서비스를 제공하는 것도 배포로 보기 때문에 서버 전체의 소스코드를 공개해야 하는 의무가 있어서 사용빈도는 낮지만 여기에 같이 적어 놓았습니다. |  |
| 22 |  | 이 10개의 라이선스도 소스코드 공개의무 강도에 따라 다시 Permissive 라이선스와 Copyleft 라이선스로 구분할 수 있습니다.  이 아이콘을 보면 대충 느낌은 오실텐데, Permissive 라이선스는 좀 쉬워 보이고, copyleft 라이선스는 뭔가 좀 어려워 보이죠? |  |
| 23 |  | Permissive 라이선스는 의무사항이 많지 않아서 상대적으로 준수하기가 쉬운 라이선스입니다.  대표적으로 MIT, Apache, BSD 라이선스가 있습니다. |  |
| 24 |  | 이 세 가지 라이선스의 의무사항은 거의 비슷합니다.  내가 이걸 사용했고 이것의 저작권자는 누구다 라는 것만 명시하면 됩니다.  Apache 라이선스에는 특허 보복 조항이 있어서 GPL 라이선스와 함께 사용할 수가 없는데 Apache와 GPL은 동시에 사용할 수 없다고 기억해 두시면 됩니다. |  |
| 25 |  | Copyleft 라이선스는 Permissive 라이선스에 비해 의무사항이 복잡하고 까다롭습니다.  Copyleft의 로그는 Copyright의 C를 뒤집어 놓은 형태입니다. |  |
| 26 |  | Copyleft 라이선스의 가장 큰 특징은 파생저작물을 배포할 때 동일한 라이선스로만 배포해야 한다는 것입니다.  BSD 같은 건 배포할 때 사용자가 라이선스를 다른 걸로 바꾸는 것이 가능합니다. 그런데, Copyleft 라이선스는 원 라이선스를 변경하는 것이 불가능합니다. 여기서는 제가 GPL 라이선스를 예로 들었는데요. 한번 GPL은 영원히 GPL로 배포할 수 밖에 없습니다. |  |
| 27 |  | 그리고, 라이선스에 따라서 사용자가 만든 파생저작물의 소스코드도 배포해야 하는 경우도 있습니다.  그러므로 소스를 공개할 마음이 없다면 copyleft 라이선스의 오픈소스를 사용하면 안 되겠죠? |  |
| 28 |  | Copyleft 계열의 주요 3개 라이선스를 살펴 볼게요.  먼저 GPL 인데요. GPL 라이선스의 오픈소스는 해당 바이너리 전체의 모든 소스코드를 공개해야 하는 의무가 있습니다. 정적 링킹이든 동적 링킹이든 그런거 따지지도 않아요. 사용하면 무조건입니다. |  |
| 29 |  | 그 다음은 LGPL인데요. 이건 GPL보다는 조금 완화된 라이선스인데요.  오픈소스를 수정해서 사용하거나 정적 링킹으로 사용하면 해당 바이너리의 오브젝트 코드를 공개해야 합니다. 그래서 소스 공개를 피하려면 원본 그대로, 동적 링킹으로만 사용해야 됩니다. |  |
| 30 |  | 그리고 제일 무시무시한 AGPL 라이선스인데요. 웹 서버 코드는 고객에게 배포하지 않잖아요. 근데 얘는 웹 서비스를 통해 서비스를 제공하는 것도 배포로 보기 때문에 서버 전체의 소스코드를 공개해야 하는 의무가 있습니다. |  |
| 31 |  | **자, 그럼 여기서 정리 한번 할게요. 이 페이지 아주 중요해요. 잘 봐주세요.**  GPL, AGPL 처럼 강한 copyleft 계열의 라이선스는 소스코드 공개 의무가 있습니다.  LGPL, EPL 처럼 약한 copyleft 계열의 라이선스는 수정해서 사용하거나 정적 링킹했을 때 오브젝트 코드 공개 의무가 있습니다.  MIT, Apache, BSD와 같은 permissive 계열의 라이선스는 대부분 사용사실과 저작권자에 대한 고지를 하면 됩니다. |  |
| 32 | **Creative Commons 라이선스** | **다음은 CCL, Creative Commons License에 대해서 설명 드리겠습니다.**  CCL은 코드는 아니고 이미지나 문서와 같은 창작물에 대해 일정한 조건 하에 다른 사람에게 이용을 허락하는 라이선스입니다.  대표적으로 우리가 위키 백과 같은데 보면 맨 아래에 크리에이티브 커먼즈-저작자 표시-동일조건변경허락과 같은 말이 있는데, 이게 바로 CCL 입니다. |  |
| 33 |  | CCL은 네 가지 라이선스 표기를 조합해서 이용 허락 조건을 만들 수 있는데요.  BY는 CCL에 반드시 포함되는 필수조항으로써 저작자의 이름, 출처를 표시해야 한다는 거구요.  NC는 Non Commercial 이라고 해서 영리 목적으로는 사용할 수 없다는 거에요. 그러니까 NC가 붙어 있으면 회사에서 사용할 수가 없어요.  ND는 변경할 수 없다. 있는 그대로만 쓰라는 거구요.  SA는 변경할 수 있는데 원 저작물과 동일한 라이선스를 유지하라는 거에요. 카피레프트 같은 거죠. |  |
| 34 | **폰트 라이선스** | **다음은 폰트 얘기인데요. 코드는 아니지만 폰트 얘기는 하고 넘어가야 할 것 같아서 넣었습니다.**  오픈소스에는 코드만 있는 것은 아니에요. 오픈 폰트로 배포되는 것도 있어서 많은 부서에서도 사용하고 계신데요. |  |
| 35 |  | 여기서 문제, PC를 지급받으면 오피스 프로그램에 맑은 고딕 폰트가 설치되어 있는데요. 과연 제한없이 사용가능한 폰트일까요? |  |
| 36 |  | 당연히 안 되니까 제가 여러분들께 여쭤 봤겠죠?  맑은 고딕은 산돌 커뮤니케이션에서 제작을 한 폰트인데 저작권을 마이크로소프트에서 가지고 있고, 그래서 [MS 오피스]에서만 사용할 수가 있습니다.  포토샵으로 포스터를 만드는데 거기에 맑은고딕체를 썼다.. 이건 라이선스 위반이 되는거구요.  윤고딕도 마찬가지에요. 얘는 우리가 별도 비용을 지불하지 않고도 한컴 오피스에서는 사용 가능해요. 근데, 다른데서 윤고딕을 사용한다? 그러면 라이선스 위반이 되는거죠. |  |
| 37 |  | 오픈폰트도 종류가 너무 많아서 일일이 찾아보기가 어려운데, 무료폰트 포털이 있으니까 오픈폰트를 사용하기 전에 여기서 라이선스를 확인해 보면 됩니다. |  |
| 38 | **학습정리** | 네. 챕터 1은 여기까지구요. 제가 말씀드린 내용 중에 꼭 기억해 주셨으면 하는 내용들만 서머리에 정리해 두었으니까 다시 잘 봐 주세요. |  |

**Chapter 2: 올바른 오픈소스 코드의 사용 (10분)**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **탭** | **강의 스크립트** | **비고** |
| 39 | **Chapter 2 제목**  **올바른 오픈소스 코드의 사용** | 자, 그럼 챕터 2, “올바른 오픈소스 코드의 사용” 에 대해 살펴보겠습니다. |  |
| 40 |  |  | 41p의 샘플 페이지입니다. |
| 41 | **라이선스 분류표** | **이것은 제가 챕터1에서 설명드렸던 내용을 다시 라이선스 분류표로 만든 것입니다.**  Permissive 라이선스에 해당하는 오픈소스는 개발할 때 자유롭게 사용하시면 되고, copyleft 라이선스에 해당하는 오픈소스는 사용하기 전에 소스공개를 해도 괜찮은지 판단해 보시기 바랍니다. 그 외 여기에 없는 오픈소스를 개발에 사용하려면 담당부서에 미리 문의를 해 주세요. 담당부서에서 검토해서 사용 가능 여부를 알려드리겠습니다. | 각 회사의 라이선스 정책대로 분류표를 만들어서 설명하세요. |
| 42 | **라이선스 확인 방법** | **자, 그러면 오픈소스의 라이선스는 어떻게 찾을 수 있는지 살펴보겠습니다.**  첫 번째로 GitHub 내 LICENSE 파일 또는 COPYING 파일에서 찾을 수 있고요.  두 번째로 소스코드 상단의 저작권 또는 LICENSE 텍스트에서도 찾을 수 있습니다.  거기서도 찾을 수가 없다면 구글에서 검색을 통해서 오픈소스 원본 저장소나 프로젝트 페이지로 가서 찾을 수도 있습니다. |  |
| 43 |  | 검색이나 다른 사람의 블로그를 통해서 오픈소스를 다운 받기도 하는데요. 이런 경우에는 소스코드의 주석을 확인하거나 개발자에게 직접 물어보는 방법도 있습니다. |  |
| 44 |  | 만약에 라이선스를 찾을 수 없다고 해서 저작권이 없는 코드라고 생각해서는 안 됩니다. 그런 경우에는 저작권자의 연락처, 예를 들어 메일 주소 같은 걸 찾아서 문의를 해 보세요. 제가 샘플로 사용할 수 있는 영문 이메일 문장을 아래에 넣어 놨는데요. 이렇게 물어보면 대부분의 개발자들은 사용해도 된다고 답변해 주거나 라이선스가 무엇인지 알려 줄 것입니다. |  |
| 45 | **오픈소스 컴플라이언스**  **프로세스** | **다음은 오픈소스 컴플라이언스에 대해 살펴보겠습니다.**  오픈소스 컴플라이언스란 오픈소스 라이선스와 관련된 리스크를 사전에 식별하고 평가하여 리스크 발생 가능성을 최소화하기 위한 활동입니다. 예를 들어, 소스코드를 검사해서 그 안에 GPL 같이 소스코드를 공개해야 되는 오픈소스가 있는지 사전에 확인을 해서 만일 그런 것이 있다면 제거하거나 다른 오픈소스로 교체를 하거나 코드 공개 준비를 하는 겁니다. 이건 빨리 하면 할수록 좋습니다. 왜냐하면 늦게 하면 늦게 할수록 나중에 수정을 하는데 비용이나 시간도 더 들어가게 되고, 만약 배포가 되어 버린 상태에서 GPL 코드가 있다는 것이 발견이 되면 일이 정말 복잡해 지거든요. |  |
| 46 |  | 전체적인 프로세스를 살펴보겠습니다. 이건 외울 필요는 없고요. 대략 이런 식으로 진행된다는 흐름만 이해하시면 됩니다.  저희가 소스코드를 받아서 스캔을 하고, 개발부서와 함께 그 결과를 크로스 체크 하게 됩니다.  그 다음엔 소스코드 공개 의무가 있는 오픈소스가 있는지 확인을 하고, 만일 그런 것이 있다면 코드 수정을 요청 드리게 됩니다. | 46~55p는 각 회사의 상황에 맞게 커스터마이징하여 설명하세요. |
| 47 |  | 그러한 과정들은 모두 기록을 해서 관리를 하고 있고요. 배포하는데 이슈가 없도록 수정이 다 끝나면 라이선스 고지문을 만들어서 프로그램을 배포할 때 함께 고지하게 됩니다. 이게 전반적인 컴플라이언스 활동의 프로세스구요. |  |
| 48 |  | 이제부터는 각 단계마다 어떤 것을 하는지 살펴보겠습니다.  오픈소스 컴플라이언스의 대상이 되는 코드는 외부로 배포되는 클라이언트 코드, 서버 코드, 웹서비스 코드입니다. 사실 웹서비스 코드는 배포되는 것은 아닌데, AGPL 오픈소스가 사용되었는지 확인을 하기 위해서 검증은 해야 됩니다. |  |
| 49 |  | 저희가 소스코드를 공유 받으면 그걸 오픈소스 식별 도구에 업로드해서 스캔 작업을 진행합니다. | 각 회사의 오픈소스 탐지 도구의 스크린샷을 보여주면서 설명하세요. |
| 50 |  | 그 다음은 저희가 라이선스 검토를 하게 되는데요. 라이선스 간 요구조건이 모순(라이선스 충돌) 되는 것은 없는지 확인을 합니다. 예를 들어서 MPL와 GPL은 라이선스의 요구조건이 모순되는 부분이 있어서 함께 사용할 수가 없습니다. |  |
| 51 |  | 그리고, AGPL, GPL, LGPL와 같이 소스코드 공개 의무가 있는 오픈소스는 없는지 확인을 하구요. |  |
| 52 |  | 만일, 그런 오픈소스의 사용이 발견되면, 코드 공개를 준비하거나 그게 싫다면 해당 코드를 제거 또는 다른 오픈소스로 대체를 합니다. 필요하다면 상용 라이선스를 구매하기도 합니다. |  |
| 53 |  | 여기까지 모든 이슈가 클리어되면 오픈소스 사용목록을 BOM(Bills of Material)에 등록을 합니다. BOM은 우리가 배포하는 소프트웨어에 어떤 오픈소스를 사용하고 있는지를 관리하는 장부입니다. | 각 회사의 BOM 관리 도구 스크린샷을 보여주면서 설명하세요. |
| 54 |  | 자, 그럼 이러한 검증은 한 번만 하면 끝나느냐…? 그건 아니고요. 프로젝트 구성이나 라이브러리 의존도가 변경되는 경우 라이선스 재검증이 필요합니다. |  |
| 55 |  |  | 56p의 샘플 페이지입니다. |
| 56 |  | **그러면, 오픈소스 사용 고지문을 어떻게 준비하고 고지하는지 그 절차에 대해 살펴보겠습니다.** | 각 회사의 상황에 맞게 커스터마이징하여 설명하세요 |
| 57 |  | 고지문에는 어떤 내용들이 포함되는지 보여 드리겠습니다.  맨 위에부터 보시면, 이 프로그램은 오픈소스를 사용했다는 것을 명시하고, 그 아래를 보면 문의사항이 있으면 어디로 문의하라는 연락처도 기재해 둡니다. 그리고 그 아래로는 사용한 오픈소스 목록들을 빠짐없이 기재하게 됩니다. 이게 고지문의 일반적인 형태이구요. |  |
| 58 |  |  | 59p의 샘플 페이지입니다. |
| 59 |  | 우리 회사 제품들의 고지 사례를 보여 드릴게요. | 각 회사의 상황에 맞게 고지문 고지 방법을 작성하여 설명하세요. |
| 60 |  | 제가 다른 회사들은 어떻게 고지를 하고 있는지 찾아보았는데요.  보통 도움말이나 고객센터, 환경설정 하위 메뉴로 가보면 오픈소스 라이선스 고지문을 확인할 수 있습니다. |  |
| 61 |  | 이거는 [배포 전 체크리스트]라고 해서 개발부서에서 소프트웨어를 배포하기 전에 체크해 보시라고 만들어 본 것입니다. 11문항 밖에 되지 않으니까 나중에 체크해 보시기 바랍니다. |  |
| 62 |  | **이제 거의 마지막인데요. 오픈소스 컴플라이언스를 준수하기 위해서 코드 작성 시 개발자 분들이 지켜주셨으면 하는 것들을 말씀 드릴게요.**  코드를 작성하면서 오픈소스를 사용할 때는 함수만 복사해서 사용하지 말고 디렉토리 단위로 사용하고, 가급적 디렉토리 구조는 원본과 동일하게 (오리지널 프로젝트명이 포함되게) 해 주세요. 일부만 카피해서 가져와서 사용하면 저희가 어떤 오픈소스가 있는지 알아내기가 어렵습니다. |  |
| 63 |  | 만일 프로젝트 구조 상 디렉토리 구조를 유지할 수 없다면, 오픈소스 프로젝트 이름이라도 확인할 수 있도록 LICENSE, README 파일은 반드시 포함시켜 주세요. |  |
| 64 |  | 마지막으로 우리 회사가 저작권을 소유하지 않은 3rd Party 코드나 데이터 파일은 가급적 별도의 디렉토리에 모아 주세요. 이렇게만 해 주셔도 저희가 오픈소스를 식별하는데 큰 도움이 됩니다. |  |
| 65 | **학습정리** | 네. 챕터 2는 여기까지구요. 코드 작성 시 준수사항은 제가 앞에서 중요하다고 말씀드렸던 만큼, 학습정리에서도 한 번 더 보여 드리겠습니다. |  |
| 66 |  | 그 밖에 오픈소스와 관련해서 문의사항이 있을 때는 OOO로 메일을 주세요.  여기까지가 제가 준비한 오픈소스 교육 내용이었구요. 끝까지 교육 내용 경청해 주셔서 감사합니다. | 오픈소스 관련 문의처와 담당자 이름을 기입하세요. |

끝.