

The Programmer's Path : Asking the Mentor

<https://product.kyobobook.co.kr/detail/S000001469857>

소프트웨어 장인 정신

- └ 성장 마인드세트 - 캐롤 드웩 : 당신을 영리하고 재능 있게 만드는 것은 노력, 실패란 단지 다음번에 다른 방법을 시도해 볼 수 있도록하는 보상.
- └ 아틀 가완디 : 당신을 둘러싼 세계로부터 얻는 피드백을 바탕으로 항상 적응하고 변화해 갈 필요성, 자신의 행위에서 적합하지 못한 면을 깨닫고 그 해결책을 찾으려는 자발성
- └ 독단적이기보다 실용적이라는 욕구, 여기에는 오늘 일을 잘 마무리하기 위해서라면 이론적인 순수함이나 앞날의 완벽함을 준비하는 일을 기꺼이 양보하는 태도
- └ 버지니아 포스트렐 : 모든 실험이나 아이디어가 다 훌륭하지는 않겠지만, 새로운 아이디어를 시험함으로써만 우리는 진짜 개선점을 발견할 수 있다. 개선의 여지란 언제나 존재한다.
- └ 내적 통제 위치, 스스로 운명을 지배하고 책임지고자 하는 태도
- └ 우리는 다름을 끌어안고 가겠다는 서약
- └ 프로세스 중심적이기보다 역량 중심적

견습생 -> 숙련공 -> 마스터

학습 상황이란 본질적으로 자기가 무엇을 하고 있는지 정말로 알고 있는 누군가를 도우면서 배워 가는 상황이다 - 크리스토퍼 알렉산더 외

언어를 하나 선택하고 그 언어에 능숙해져라

멘토를 찾아라

마음 맞는 사람들

특정한 기술 분야에 집착하지 말고, 개별 상황에 알맞은 해법을 고를 수 있도록 광범위한 배경 지식과 경험을 충분히 쌓아두어야 한다 - 데이브 토머스와 앤디 헌트

대개 발걸음 하나하나마다 새로 시작한다는 느낌이 들어야한다. 이것이 초심이며 "되고 있음"의 상태이다 - 순류 스즈키

위로 오르기 위해서는 당신이 이미 잘하는 것을 내려놓아야 한다. 그리고 골짜기로 미끄러져 내리기도 하면서 단단히 디디고 선 곳을 떠나야 한다. - 제리 와인버그

특정 프로그래밍 패러다임으로 작성한 프로그램을 하나 골라서 다른 패러다임에 속한 언어로 다시 구현해보기

전염되는 의욕

지식을 가진 것, 그리고 그 지식을 써서 소프트웨어를 만들어 내는 역량과 실무 능력을 갖춘 것은 다르다. 여기에 장인장신의 역할이 있다. - 피트 맥브린

구체적인 기술을 습득해서 유지하라

내일 나는 더 어리석게 보일 필요가 있으며, 거기에 대한 느낌도 더 나아질 것이다. 가만히 있으면서 일이 어떻게 돌아가나 살피는 것은 별로 효과가 없는 것 같다. - 제이크 스크럭스

무참한 실채를 맛본 적이 한 번도 없다면, 당신은 뭔가 가치 있는 일을 시도했던 적인 한 번도 없었다고 해야 한다. - 크리스토퍼 호킨스

깊은 쪽으로 뛰어들어라

당신이 가려는 곳을 바라본 다음에 지금 어디쯤 있는지를 보면 항상 터무니없다는 생각이 들 겁니다. 그리고 나서 당신이 걸어왔던 길을 보고 있노라면 그 속에서 어떤 패턴 같은 것이 드러날 거예요. 그 패턴으로 당신의 앞길을 비추어 본다면, 가끔은 그 무엇인가를 찾아낼 수 있을 겁니다. - 로버틀 퍼식

"합기도를 마스터하려면 얼마나 오래 걸려요?" "자네 얼마나 오래 살 것 같나?" - 조지 레너드

정말로 프로그래밍을 잘 하게 된다는 것은 일생의 과업이며, 지속적인 배움과 연습으로 이루어가는 대담한 계획이다. - 론 제프리즈

달인의 경지로 이르는 길을 한 걸음 딛을 때마다 당신의 목적지는 두 걸음 멀어진다. 그 경지는 평생의 노력이 필요함을 받아들이고, 거기에 이르는 여정을 사랑하는 법을 배워라. - 조지 레너드

길게 보아라

자신만의 지도

정확한 자기 평가

장인 정신은 튼튼한 상호 관계에 기초

프로그래머가 일하는 모습을 본 적이 있는 사람이라면, 자기 방식대로 일할 수 있는 기회가 주어졌을 때 프로그래밍은 그 자체로 가장 큰 동기를 부여한다는 것을 알 것이다. - 제럴드 와인버그

지속 가능한 동기 부여와 직위

소비 제일과 임시 변통이 판치는 이 사회의 유혹에 넘어가서 우리는 가끔 잘못된 길을 선택하는데 거기서는 단지 성취의 환상, 만족의 그림자를 얻을 뿐이다. - 조지 레너드

당신이 가는 길로 가지 않는다고 해서 그 사람들이 길을 잃은 것은 아니다. - 잭슨 브라운 주니어

예술보다는 기예에 집중

여우의 머리가 되기보다는 사자의 꼬리가 되어라

연관된 문제를 풀기 위해 힘을 모으고 있는 인재들의 커뮤니티보다 강력한 것은 없다 - 폴 그레이엄

배운것 기록하기

바닥을 쓸어라

능력의 폭 넓히기

부숴도 괜찮은 장난감과 소스 활용

연습, 연습 또 연습

일하면서 성찰하라

배운 것을 기록하고 공유

피드백 루프 만들기

실패하는 법 배우기

모르는 것을 배우는 일이, 어떻게 하는지 이미 아는 일을 행하는 것보다 더 중요할 때가 흔히 있다. - 짐 하이스미스

구글 리더, 알파 깁

트위터의 소프트웨어 전문가 팔로우

온라인 메일링 리스트 가입

최근 만들어진 로컬 유저 그룹에 참여

고용주에게 기술 컨퍼런스 제공 설득

책 읽은 후 저자에게 연락

온라인 대학 강좌나 팟캐스트, 동영상

우리가 마스터라고 알고 있는 사람들은 어떤 특정한 기술에 더 능숙해질 목적으로 거기 전념 하지는 않는다. 사실 그 사람들은 연습하는 것 자체로 좋아하며, 이 때문에 그들이 점점 더 능숙해지는 것이다. 그리고 더 능숙해질수록 기본적인 동작을 더 즐기게 되는 선순환이 완성한다. - 조지 레너드

의도적 수련

인위적인 제한에 걸려 있을 때, 우리는 자신의 능력을 한계까지 밀어붙이는 '장난감' 프로그램을 만들고 써 종종 경험을 볼 수 있다. - 도널드 커누스

개인용 위키와 외부용 위키

프로그래머가 되고자 준비한 가장 좋은 방법은 프로그램을 짜는 것이며, 다른 사람들이 작성한 뛰어나 프로그램을 공부하는 것이다. - 빌 게이츠

다른 사람의 코드를 찾아서 읽어라

패턴, 관용 어법, 우수한 사례들에 대해 배우는 가장 좋은 방법은 오픈소스 코드를 읽는 것입니다. - 크리스 웜스트拉斯

자기 성찰은 쉽지 않은 일일지만, 우리는 자신의 성공보다 실패를 연구함으로써 더 많은 것을 배울 수 있다고 나는 믿는다. - 노엄 커스

쓰는 행위 자체가 가진 힘 역시 과소평가해서는 안된다. - 아쿨 가완디

개척자로서 우리에게는 광야를 지날 때 뚜렷이 보이는 안전한 길을 남겨야하는 부가적인 책임이 있다. -

우리 소프트웨어 업계 종사자들은 어느 정도는 눈에 보이지 않는 제품을 가지고 일하지만, 눈에 보이지 않는 점 때문에 피드백에 대한 필요성은 오히려 커진다. - 노엄 커스

배우는 사람은 자기만의 학습 계획을 스스로 세운다. - 하워드 S. 베커

좋은 프로그래밍 책을 두 달에 한권, 즉 일주일에 대략 35페이지정도만 읽어도 단신은 이내 이분야에 대해서 확실한 감을 갖게 될 것이며 주변의 거의 모든 이들과 구별되는 수준으로 올라설 것이다. - 스티브 맥코넬

당신의 직업이나 관심 분야에서 위대한 저작들, 즉 지금까지 있었던 가장 훌륭한 책과 기사, 연설문 등을 찾아서 그것을 진지하게 공부하기 시작하라 - 조슈아 캐리에브스키

바퀴 자국 안에 빠지게 되면, 아무리 바퀴를 돌려도 계속 제자리고 유일한 진척이라고는 더 깊게 파이는 자국뿐이다. 흡은 다르다. 바퀴가 흡에 맞물리면 당신은 힘들이지 않고 앞으로 전진한다. - 트와일라 터프

극적으로 변화하는 시기에 미래를 계승하는 것은 배우는 자들이다. - 에릭 호퍼