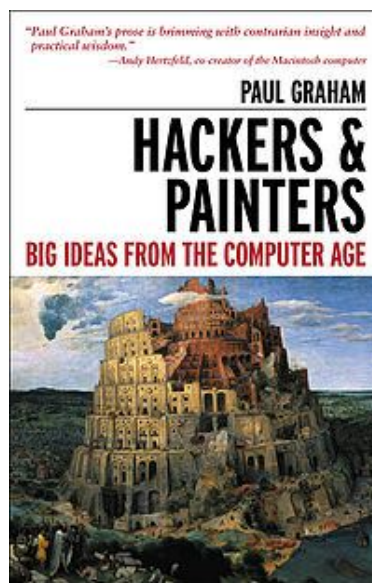


# 폴 그레이엄의 해커와 화가 Hackers and Painters (3/5)

PRODUCT

9월 6, 2013 박상민

**Editor's note:** 프로그래머이자 해커인 박상민님이 폴 그레이엄 Paul Graham의 에세이 '해커와 화가' 번역을 총 5회에 걸쳐 게재합니다. Y combinator를 창업한 폴 그레이엄은 Dropbox, Reddit, Airbnb등의 스타트업을 키워낸 대가로, 투자자이면서도 뛰어난 프로그래머이며 수필가로도 명성을 떨치고 있습니다. (원문 보기)



해커는 과학자라기 보다는 maker이기 때문에, 영감을 얻을 수 있는 분야는 과학보다는 다른 종류의 maker일 것이다. 그 관점에서 미술보다 해킹에 대해 더 잘 드러낼 분야가 또 있을까?

우리가 미술에서 배울 수 있는것 아니 최소한 확인할 수 있는 것은 어떻게 해킹을 처음 배우는가 하는 점이다. 당신은 그림을 직접 그려가며 미술을 배운다. 덩동댕! 해킹 역시 그렇다. 대부분 해커들은 대학에서 프로그래밍 과목을 수강한후 해킹을 배우지 않는다. 13살때쯤 스스로 프로그램을 만들어보며 해킹을 배운다. 사실 대학 수업역시 진짜 만들어보면서 배우기 마련이다.

화가들은 자신만의 독특한 흔적을 남기기 때문에, 그림을 그려나가며 배웠다는 사실을 알수 있다. 어떤 화가의 작품을 시간순으로 나열해 보면 각각의 작품들이 그 전의 그림들 토대 위에서 만

들어진것을 볼 수 있다. 회화에서 특별히 뛰어나게 잘 표현된 부분이 있다면 그 표현의 버전 1을 그 전의 그림들에서 발견할 수 있을 것이다.

대부분의 maker들이 이런 방식으로 일한다. 작가와 건축가 역시 마찬가지다. 아마도 해커는 화가와 비슷한 방식으로, 한가지 프로젝트에 몇년간 계속해서 일하기 보다는 자주 처음부터 프로젝트를 새로 시작하면서 기존의 아이디어를 통합해서 발전시켜나가는 것이 좋을 것이다.

해커들은 직접 코딩하면서 배운다는 사실이 바로 해킹과 과학이 별개의 것이라는 증명이된다. 과학자들은 과학을 랩과 문제풀이를 통해 배운다. 과학자들은 처음 배울때 완벽한 것에서 시작한다. 이미 잘 증명된 문제를 실험으로 재현하며 배우고 훗날 자신의 이론을 세우고 실험하는 단계에 이른다. 반면 해커는 처음 시작부터 자신의 일을 한다. 그냥 아직 아주 어설픈 뿐이다. 해커들은 처음에 독창적인 것을 하고, 시간이 갈수록 더 나아지는 반면 과학자들은 처음에 완전한 것에서 시작하고 시간이 갈수록 독창성을 가미한다.

Maker들이 배우는 또 다른 방법은 예제를 통해서이다. 화가에게 있어 미술관은 기교를 참조하는 라이브러리다. 수백년간 지속되온 전통적인 미술 교습법은 거장의 작품들을 모방하는 것인데, 그 이유는 모방을 통해서 작품이 만들어진 방식을 유심히 관찰하게 되기 때문이다.

작가역시 그렇다. 벤자민 프랭클린은 Addison과 Steele이 쓴 수필들을 요약해가면서 글쓰는 법을 배웠다. Raymond Chandler역시 같은 방식으로 탐정 소설을 배웠다.

해커 역시 마찬가지로 훌륭한 프로그램을 참조하면서 프로그래밍을 배운다. 단순히 프로그램의 돌아가는 것을 보는것 뿐 아니라 소스 코드를 참조하면서 말이다. 오픈소스 운동의 덜 알려진 혜택중 하나는 프로그램을 배우기 쉽게 한다는 점이다. 내가 처음 프로그램을 배울때는 대부분 책에 소개된 예제를 통해서였다. 그 당시 예제중 가장 큰 소스 코드는 Unix였는데, 그땐 오픈소스가 아니었다. 소스코드를 읽은 대부분 사람들은 John Lions의 책을 복사한 해적판을 읽었는데, 1977년에 쓰여진 그 책은 1996년까지 출판이 금지되었었다.

미술을 통해 얻을 수 있는 또 다른 예제는 점진적인 개선을 통해 작품을 창조해가는 점이다. 그림은 대부분 스케치에서 시작한다. 구체적인 것은 점차 채워져 나간다. 하지만 그 과정이 단지 채워나가는 것만이 아니다. 때때로 처음 계획은 실수로 판가름난다. xray를 통해 보면 수도 없이 많은 그림들이 테두리가 옮겨져있거나 표면 형태들이 새로 적용된 것을 볼 수 있다.

그것이 바로 미술에서 배울 수 있는 점이다. 해킹또한 이렇게 이루어져야 한다고 생각한다. 프로그램의 스펙이 완벽하다고 기대하는건 비현실적이다. 처음부터 그 사실을 인정하고, 프로그램을

만들며 스펙을 고쳐나가는 것이 훨씬 낫다.

(큰 회사의 구조는 이것이 힘들게 되어있는데, 이것이 스타트업의 잇점중 하나다.)

아마 지금쯤 대부분 사람들은 미성숙한 최적화 (premature optimization)의 위험에 대해 알 것이다. 내 생각에 우리는 미성숙한 디자인 (프로그램이 해야하는 일을 너무 일찍 결정하는 것)에 대해서 역시 걱정해야 한다.

올바른 툴은 이런 위험을 피하는데 도움을 줄 수 있다. 좋은 프로그래밍 언어는 유화가 그러하듯이 해커가 마음을 쉽게 바꿀수 있게끔 도와야 한다. 그점에서 동적 타이핑이 승자라고 할 수 있는데 처음부터 특정한 데이터 표현에 묶이지 않기 때문이다. 하지만 그러한 유연성의 핵심은 언어를 아주 추상적으로 만드는 것이라고 생각한다. 바꾸기 가장 쉬운 프로그램은 아주 짧은 것이다 (역: 언어가 추상적일수록 소스코드가 짧다).

(다음 편에 계속)



**박상민**

Ph.D. in Computer Science, University of Virginia 현재 워싱턴주 시애틀 거주 (Bellevue)하고 있는 소프트웨어 엔지니어(Eucalyptus systems) 이며, 해커, 오픈소스 팬, 블로거(Hacker, Open source enthusiast, and writer)이자 예쁜 두 딸의 아빠 (Father of two lovely girls)



# KOREATECHDESK.COM

한국 스타트업 영문 기사 전문 미디어 - 코리아테크데스크  
PRESS@KOREATECHDESK.COM



*Be the First to Comment!*

B I U {} [+]



0 COMMENTS



정보 **개인정보처리방침** 청소년보호정책

제보 : press@besuccess.com | 광고 : info@besuccess.com

주소 : 07335 서울시 영등포구 여의대로 108 파크원 타워1, 5층 535호 | #535, 5F, Parc1 Tower1, 108, Yeoui-daero, Yeongdeungpo-gu, Seoul, Korea 07335

주식회사 비석세스 | 제호 : 비석세스 | 등록번호 : 서울, 아01147 | 등록일자 : 2010년 02월 16일 | 발행일자 : 2011년 05월 17일 | 전화번호 : 070-4681-0074

발행인 : 정현욱 | 편집인 : 전진주 | 청소년보호책임자 : 전진주