

[Std.IO] Hello, World! 컨퍼런스 정리

-by 서 동현

행사 일정표

시간	Activity	진행자	Contents
9:30 - 10:00 (30분)	참가자 등록	Std.IO	
10:00 - 10:20 (20분)	행사 소개 및 Std.IO 소개	김혜원, 조광희	
10:20~12:00 (100분)	세션 1	김종찬 Software Maestro	컵공과 학생으로써, 학생을 넘어서기
		김창준 Agile Consulting	지금 알고 있는 걸, 그때도 알았더라면
12:00~13:00 (60분)	Lunch Time		
13:00~14:40 (100분)	세션2	양찬석 Google	안드로이드 어플리케이션 개발 올바르게 시작하기
		임성현 스펙트라	사례로 풀어보는 졸업작품 기획
14:40~15:20 (40분)	Breaking Time	안혜미, 임원균	Mini Quiz & Event
15:20~17:00 (100분)	세션3	유명환 예세스 주식회사	안드로이드는 OS가 아니라 플랫폼이다!
		권순선 Google	오픈소스 & 커뮤니티와 함께 성장하기
17:00~17:15 (15분)	경품 추첨	Std.IO	경품 추첨 Event



Hello, World! 는 그 동안 현업 개발자 분들께 초점이 맞추어져 있던 기존의 개발자 행사에 아쉬움을 느낀 대학생들이 기획 , 준비하고 있는 학생의 , 학생에 의한 , 학생을 위한 개발자 행사 입니다.

최근 대학생 스타트업 , 얼마전 폐지된(..) 대학생 산업기능요원 등 IT 업계에서 학생 개발자들의 활동이 적지 않다고 생각됩니다. 개발자를 꿈꾸고 노력하고 있는 학생들은 여러 개발 관련 행사에 참가합니다. 하지만 많은 개발자 컨퍼런스들은 기술적인 내용 또는 학생의 초점에서 벗어난 주제를 가지고 이야기하는 경우가 많기 때문에 정작 학생이기에 가지고 있는 의문이나 고민을 이야기해주지는 않았습니다.

Hello, World! 는 학생 개발자들에게 기술적인 내용 뿐만 아니라 미래 설계에 필요한 정보를 제공해 자기 계발에 도움이 되게 하자는 취지를 가지고 있습니다.

학생 개발자들에게 필요한 좋은 이야기 , 좋은 경험을 전달해주시실 멘토 개발자 분들을 Speaker 로 모셨습니다. 개발자로서의 진로 그리고 미래에 대해서 궁금한 점이 있으신 분들은 이번 행사에 참여하셔서 선배 개발자들의 진솔한 이야기를 들어 보실 수 있는 좋은 경험이 되리라고 생각합니다.

#STD.IO>

#STD.IO>

#STD.IO>

#STD.IO>

#STD.IO>



김종찬

- 송실대학교 컴퓨터학부 2학년
- Software Maestro 4기 연수생
- GDG Soongsil University Founder

< 컴공과 학생으로써, 학생을 넘어서기 >

"스스로 생각하고 움직이는 학생은 그 누구보다 학생 그 자체를 뛰어넘을수 있다고 생각합니다."

어렸을적 가장 행복했던시간은 컴퓨터를 잡고 보내는 시간, 그로인해 한국디지털미디어고등학교 라는 IT특성화 고등학교에 들어가고, 재수라는 시련끝에 송실대학교 컴퓨터학부에서 수학 할 수 있게 되었습니다. Google의 Support를 받아 한국 최초로 대학교 GDG를 창설하고, 그곳에서 개발자들이 어떤 것이지, 그들이 모여면 어떤 시너지를 낼 수 있는지 경험으로써 터득하게 되었습니다. 그리고 개발자로서 더 넓은 경험을 하기위해 Software Maestro에 도전하게 되었습니다. 현재는 1단계를 넘어 2단계 연수생으로써 팀 프로젝트를 구성하기 위해 고군분투하는 중입니다. 어렸을적부터 이 분야에 깊게 관심을 가져왔고 욕심을 냈던 그저 그런 학생중 한명으로써, 그냥 그저그런 학생의 틀을 뛰어넘기위해 제가 경험해왔고, 도전해왔던 일들을 소개해드리려 합니다. 그리고 이후에는 "학생", "학생개발자"라는 우리들이 함께 그 틀을 뛰어넘기위해서는 어떤 의식과 액션이 필요한지에 대해서 제 나름대로의 생각을 공유하고, 이후에 우리가 어떻게 함께 움직일수 있을지에대해서 이야기 해보려 합니다.

#STD.IO>

#STD.IO>

#STD.IO>

#STD.IO>

#STD.IO>



김창준

- Agile 컨설팅 대표
- Software Maestro Mentor

< 지금 알고 있는 걸 그때도 알았더라면 >

"인간적이면서 생산적인 길 찾기, 나날이 자라기, 나와 주변의 긍정적 변화"

애자일 컨설팅 대표입니다. 2000년경 국내에 애자일 소프트웨어 개발 방법론을 소개했습니다. 현재 조직과 개인이 변화하는 걸 도와주고 있고 "애자일 이야기"(<http://agile.egloos.com>)라는 블로그에 글을 쓰고 있습니다.

소프트웨어 개발과 관련된 일을 하는 사람들이 지금 알고 있는 걸 그때(대학생 때)도 알았더라면 하는 건 뭘까요? 소프트웨어 개발에 관심이 있고 앞으로 관련 일을 하고 싶은 사람들이 현재 가치있게 시간을 쓰기 위해서는 어떻게 하루를 살아가야 할까요? 여기에 대해 여러분들과 함께 불가에 앉아 편안한 이야기를 나눠 봅시다.



양찬석

- 구글 코리아
- Android Developer Program Engineer

< 안드로이드 어플리케이션 개발
올바르게 시작하기 >

"소프트웨어 개발이 즐겁다면 개발자의 삶도 나쁘지 않습니다."

초등학교 시절, 컴퓨터 게임과 사랑에 빠진 후 한결같이 소프트웨어 개발자를 꿈꿔왔습니다. 그리고 2014년 새해가 밝으면서 소프트웨어 개발자로 일해온지 정확히 10년 차가 되었네요. 그동안 삼성전자 연구소에서 미디어 셰어링에 관한 프로젝트에 참고했고, 관심사 기반 커뮤니티 빙글에서 안드로이드 개발자로 일했고, 작년 하반기 부터 구글 코리아에서 개발자 분들을 위해 개발을 하는 Android Developer Program Engineer 로 근무하고 있습니다. 돌이켜 생각해보면, 많은 분들의 도움과 인연 덕분에 아직까지 즐겁게 개발자 경력을 이어갈 수 있는 것 같습니다.. 앞으로는 저 역시 다른 분들께 좋은 인연이 될 수 있도록 조금 더 노력하고 가능한 백발이 성성할 때까지 개발 일에 발을 담그고 싶습니다.



임성현

- 스펙트라 (문자상담 솔루션 회사)
- 전자계산조직응용 기술사
- 한이음 IT멘토링 멘토

< 사례로 풀어보는 졸업작품 기획 >

"학창시절에 가능한 많은 사람들을 만나서 이야기 나누는 게 좋아요."

15년차 자바 개발자입니다. 전산실, SI개발자, PM, 솔루션 기획부터 영업까지 두루두루 경험해보았습니다. 지금은 솔루션 회사에서 제품 기획과 소프트웨어 품질관리를 담당하고 있으며, 행복한 개발자가 꿈입니다. 8년째 한이음 IT멘토링을 통해 대학생들을 만나고 있으며, 시각장애인을 위한 앱 개발등을 통해 학생들이 후배 개발자로 성장해 나가는 과정을 돕고 있습니다. 한이음 IT멘토링 이외에도 삼성 직업 멘토링, NC소프트 주관의 WoC 밥멘토, 전국 소년원의 정보화 역기능 강사로 활동해왔으며, 작년에는 프로젝트 기획 단계를 지도하는 참여자 1,000명 규모의 한이음 BIG 멘토링을 기획하고 진행했습니다. 2013년 가을에 대학 교재용으로 Practical Software Engineering을 집필했습니다.

#STD.IO> #STD.IO> #STD.IO> #STD.IO> #STD.IO>

#STD.IO> #STD.IO>

#STD.IO> #STD.IO> #STD.IO> #STD.IO>



유명환

- 엑세스 주식회사 연구소장
- 오픈소스 포럼 포럼지기

< 안드로이드는
OS가 아니고 플랫폼이다! >

"재미있는 것과 쉬운 건 별개의 이야기입니다. 사실 진짜 재미있는 일들은 어려운 문제들을 풀었을 때가 가장 재미있죠."

응답하라 1994 주인공 학번(?)으로 인터넷의 자유로움에 빠져 HTML 책 대신 브라우저 소스 코드 보기로 웹 프로그래밍을 시작하고, Hotjava 브라우저로 Java를 알게 되어 결국 Java 프로그래머로 사회 첫발을 내딛었으나, 네트워크 장비에서 돌아가던 소프트웨어의 두근거림 때문에 그 모든 것들을 뒤로 하고 임베디드 소프트웨어와 하드웨어 개발에 폭 빠져 현재까지 그 길을 걷고 있는 1년차 연구소장입니다.

10여년 간 프리랜서와 창업의 길을 걸어오면서 겪었던 개발과는 또다른 사업의 길 덕분에 개발자로선 겪기 어려운 경험들을 조금 일찍 접하면서 자연스럽게 프로젝트를 설계하고 추진하는 일이 몸에 배어 현재는 후배 양성에 무척 관심이 많으며, 그간의 경험들을 토대로 임베디드 리눅스 관련 서적을 한권 집필했고 요즘엔 클라우드와 임베디드를 융합한 ARM 서버 개발에 매진하고 있습니다.

#STD.IO> #STD.IO> #STD.IO> #STD.IO> #STD.IO>

#STD.IO> #STD.IO>

#STD.IO> #STD.IO> #STD.IO> #STD.IO>



권순선

- Google DevRel Program Manager
- KLDP Founder

< 오픈소스 & 커뮤니티와 함께 성장하기 >

" Change the world for Creative minds! "

1999년부터 사회생활을 시작하여 삼성전자에서 9년, 네이버에서 3년 8개월간 일했고 지금은 구글에서 다양한 구글 기술들을 활용한 유용한 애플리케이션/서비스들이 한국에서 많이 나오도록 하기 위한 일을 하고 있습니다. 1996년에 리눅스/오픈소스 커뮤니티인 KLDP를 만들어서 지금까지 운영하며 즐겁고도 의미있는 시간을 보냈고, 히말라야의 해발 5540미터 지점까지 올라가서 에베레스트 산을 직접 보고 온 것을 누구에게나 자랑하고 싶어하는 평범한 직장인입니다.

세션 1-1 [컴공과 학생으로써, 학생을 넘어서기 - 김종찬]

Start-Up 9

1. Computer Science 분야에 과대한 관심과 애정
 2. 작은 것, 시키는 것부터 천천히
 3. 커리큘럼이라는 울타리를 벗어나보자!!
 4. 되돌아보고 비교하며 반성하자
 5. 천천히 급하지 않게 배우며, 달라붙자
 6. 함께가자.
 7. 벽을 넘어 도전하자.
 8. 공개하고 공유하고 이끌어보자
 9. 실전, 그리고 경험.
0. 좋은 동료! => 제일 중요!

세션 1-2 [지금 알고 있는 걸, 그때도 알았더라면 - 김창준]

방법론?

일하는 방법. 그림을 그리는 과정. 현업으로 그림을 그리면 어떻게 그리는 과정.

팀프로젝트 유사점들....

굉장히 잘하는 친구, 복학생, 마음이 떠난 친구, 마음은 있는데 맘대로 잘 안되는 친구

성적이 가장 덜 중요했다는 생각

내가 했던 선택의 반복 => 열심히 했으면 직장 가서도 계속 그런 연속

****팀프로젝트에 있었던 일이 직장가서도 반복된다!

프로젝트를 했던 성적보다 그 과정들이 더 중요하다. 이게 후에 미치는 영향!

팀프로젝트, 과제를 할때 내가 했던 선택들을 잘 돌이켜봐라, 나중에 직장가서도 그 습관들이 그대로 반영된다. 학교에서 그 습관을 완성해야한다. 학교가 더 안정적이다. 새로운 실험, 다른 사람들 관찰.

나중에 제대로 할거야~ 이런건 의미가 없다.

Agile 방법론은 언제 접했는가/접한후의 변화?

우리가 무엇을 정상으로 보는가, 우리가 무엇을 비정상으로 보는가가 영향을 미친다.

설계를 제대로 하는게 정상인가?

설계를 안하는게 정산인것인가?

인간성과 생산성을 동시에 높여야 한다.

커뮤니티가 얼마나 건강한가?

저 사람은 프로젝트에 즐거운가, 교감...

자기자신의 변화 / 남에게 좋은 영향을 미치는 방법은?

변화는 쉽다, 변화의 유지가 어렵다

인간의 뇌는 경험에 의해 변화된다.

책을 보거나 말을 듣는 것보다 직접적 경험이 더 영향이 크다.

새로운 정보를 얻으면 될 수 있으면 빨리 사용하자

멘탈 시뮬레이션 - 뇌는 실제 경험과 상상을 잘 구분하지 못한다

좋은 정보를 얻었을 때, 빨리 실행하고, 이어질 수 있게 만든다. => 지속

남에게 좋은 영향을 미치는 것은... 2차적이다.

그 사람을 어떻게 살피는게 중요하다.

타인의 이해가 중요하다.

자신의 경험 설계가 중요하다.

타인을 이해하는 것에 남녀의 차이는 동기의 차이에 있다.

좋은 책과 좋은 말을 들으면 멘탈 시뮬레이션을 하라!

대학이 필요한가?

고민들은 뒤돌아 보면 아무렇지도 않다. 부정적 효과가 적다.
인간은 적응하는 동물이다. 쾌락 적응, 고통 적응

과도한 고민을 하지 않아도 된다.
허나 사람들은 안해본 것에 대한 후회가 많다.

****고민할 시간에 해라.

강사(본인)에게 있어 대학의 의미는 도서관 같았다.

위대한 사람들의 말을 직접 들은게 엄청 좋았었다.

Agile을 어떻게 잘 도입할 수 있는가?

여성이 불리하다. 자신의 편견이 가장 불리하다. 자신의 편견 극복이 중요하다

신뢰라는 면에서 이야기 하고 싶다.

성공했던 경우 : 그 사람의 정치적 파워가 좋았다. 신뢰가 높았다.
실패했던 경우 : 신뢰가 적기에 사람들이 믿지를 않는다.

신뢰가 없으면 어떤 행동을 해도 악의의 행동으로 보인다.
****따라서 신뢰를 먼저 쌓아라.

신뢰는 3가지가 있다, => 어떤 방법론을 적용해도 신뢰가 있으면 된다!
역량에 대한 신뢰, 의사소통에 대한 신뢰, 거래에 대한 신뢰

역량에 대한 신뢰 : 일을 잘할 거라는 신뢰
의사소통의 신뢰 : 그 사람이 솔직하게 말해줄 것이라는 신뢰 - 솔직하게 말할 것과 그 사람의 신뢰에 대한 비난을 말아야한다.
거래에 대한 신뢰 : 약속을 지켜라. 기한등..

정리해보면.....

**X를 하냐 안하냐는 중요치않다. 고민을 하기보다 될 수 있으면 해봐라.

**경험. 새로운 것. 경험 설계를 해봐라.

**신뢰. 신뢰에 대해 생각해봐라.

**학교에서 프로젝트 했던 것들, 그거 직장에서도 일어난다.

**도서관의 중요성!

**물어보는 것의 중요성. - 잘 보내는 것보다 많이 보내는 것이 중요하다,
확률을 높이는 것보다 샘플을 높여라.
궁금한 것이 있으면 용기를 내고 그 횟수를 늘여라.

세션 2-1 [안드로이드 어플리케이션 개발 올바르게 시작하기 - 양찬석]

왜 앱을 만들고 싶으신가요?

지도와 나침판
가능한 명확한 목표가 필요하다!

꼭 해결하고 싶은 문제
구체적이고 직접적인 보상
외부의 압력

우선 필요한 것

한잔의 커피~ (JAVA) + 달콤한 시럽? (디자인 패턴 + 프로세스 / 스레드 + 자료구조 + 리눅스)

DAC - <http://developer.android.com>

안드로이드 앱 개발자가 알아야하는 거의 모든 것
살아있는 문서! 매우 자주 업데이트 함

Design => 알아야한다! : 안드로이드 멀티스크린, 디자인 패턴(UX)
Develop => Reference : 매뉴얼, API필요할 때 찾아보면 하면 된다
***Training : 구글에서 중요하게 생각하고 잘 만들고 있다.
처음 시작할 때 이곳을 배우는게 좋다!
동기화 문제등 Step by step 으로 잘 나와있다.
철저하게 검수와 리뷰를 통해 만들어진 엄선 자료!
(단, 영어지만 영어 잘 못해도 문서는 읽을 수준은 되
되어야 한다!)

API Guide : 소설 썸으로 생각하면된다. 작동 원리, 특징, 내부적으로 어떻게 되어있나 등을 확인 할 수 있다.

Distribute

요즘은 개발자가 앱을 배포도 하기 때문에 알아두어야 좋다.

요즘 트렌드는 앱의 최소화 부분만 만들고 배포 후에 사용자의 피드백을 통해 앱을 발전 시켜나간다.

앱은 내가 만들고 싶어서 만들지만 사용자와 함께 쓰는 것이다.

앱에다 코드를 심어서 사용자의 패턴이 날라오게 만든다.

Data Analysis를 통해 사용자의 패턴(데이터)를 분석해서 해당되는 부분을 개선한다.

스택 오버 플로우 (매우 중요!)

Ask questions, get answers, no distractions

당신이 질문하고 싶은 99%의 질문은 이미 다른 누군가 이미 물어봤을 것이다.
아닌 경우는 당신의 코드의 오류일 확률이 높다!

올바르게 질문하기

질문에 연관된 제목을 달아라
상황을 설명하기
코드를 보여라!
결과도 보여야 한다
자신의 노력을 어필하라
답변에 감사하라

사각형은 중요합니다.

검색이 중요하다!!!!



!!가능한 영어로 검색하라

!!가능한 구글로 검색하라

오픈 소스 활용하기

Do not reinvent the wheel.

github (git : 버전 관리 툴)에서 만든 오픈소스 커뮤니티
android open source project : 안드로이드는 오픈소스이기에 장점

냄비 속 개구리

안드로이드는 너무 빨리바뀐다. 업데이트가 빠르다.
주변을 살피지 않으면 많은 것의 변화를 모른다.

새로운 뉴스에 귀를 기울이지 않으면 뒤쳐질 수 있다.
커뮤니티 활동을 하여야한다.

커뮤니티 & Newsletter

GDG Korea Android
+AndroidDevelopers (google+)
AndroidWeekly

Do!

- 1.DAC 접속하기
- 2,StackOvderflow 회원 가입하기
3. 개발자 커뮤니티 회원 가입하기

세션 2-2 [사례로 풀어보는 졸업작품 기획 - 임성현]

<http://spectra.clbnow.com>

한이음 IT 멘토링

오늘 할 이야기

지난 8년간 진행한 프로젝트 중 5개 선별하여 소개합니다.
졸업 작품에 대한 관점을 바꿔봅시다.
현업에서 원하는 개발자 모습도 함께 이야기 합니다.
그래서, 누가 쓸지 생각하고 결론은 고객을 알자.

살아 남는 회사는 고객이 누구인지 아는 회사가 살아남는다.

사례 1: 누가 사용할지 생각하자

어드벤처형 여행가이드 앱
: 아이들 답사 경험을 해주어야하는 아빠의 입장

사례 2 : 몸으로 뛰어보는 졸업 작품

분교/낙도 어린이에게 프로그래밍 선생님 되어보기
: IT학과 재능나눔으로 분교 학생을 위한 아두이노 가르치기

사례 3 : 50명짜리 작품 만들기

시각장애 어린이를 위한 구현동화 앱

사례 4 : 진지한 사용자 중심의 작품

시각장애자를 위한 지폐인식 앱
: 시각장애자를 위한을 빼달라 요청, 사용자가 부끄럽지 않게 만들라

***누가 쓸지 "한번 더 생각"하고 작품을 기획해보았으면.

목적으로 제대로 가고 있는지 어떻게 알 수 있나요?
실제 그 대상자를 만나봐라!

funfun.yoo@gmail.com

운영체제와 플랫폼?

Platform = CPU + OS + TOOL

Android

= ARM/x86 + Linux + Ubuntu/Eclipse

Linux in Android =>

System.img , data.img 을 제외하고 나머지는 리눅스 시스템,
ramdisk.img 은 리눅스 시스템을 램에다 올려놓고 쓴다.(변경이 없기 때문)

안드로이드에서 우분투 외 다른 운영체제에서는 안되나?

플랫폼의 정의

결국 "플랫폼"이란 "개발 상의 제약"을 뜻함!!

**제약이 있음에도 플랫폼을 채택하는 이유는?

=> 편하니까, 싸니까(인건비) // 개발기간 단축!!!!

=> 킬러앱이 없으면 그 플랫폼은 죽는다

=> 킬러앱이 나올려면 환경을 만들어줘야한다.

왜 플랫폼을 알아야 할까?

개발의 시작이 바로 "플랫폼"이다!

개발자가 반드시 알아야 할 가장 기본 중의 기본!

수없이 쏟아져 나오는 신 기술들에 대한 분석 기준!

** '개발자' 와 '직장인' 을 구분하자.

개발자 : 무엇을 발전 시키려는 사람

직장인 : 돈받을려고 일하는 사람

우리는 플랫폼에 대해 얼마나 알고 있을까?

왜 프로그램은 반드시 '메모리'에 올린 이후에 실행이 되는 걸까?

가상 메모리는 왜 사용하는 것일까?

C언어 전역변수는 초기화하지 않으면 왜 '0'으로 초기화되는 걸까?

왜 C언어에서는 사용하지 말라는 GOTO문이 있는 걸까?

어플리케이션 작성 시 스택(Stack) 사이즈를 할당해 본 적이 있나?

인터럽트 서비스 루틴은 소프트웨어 호출에 의해서 실행되는 걸까?

어플리케이션에서 사용할 수 있는 함수의 개수는 총 몇 개일까?

안드로이드 앱에 왜 main 메소드가 없을까?

안드로이드 앱의 R.java를 왜 건드리지 말라고 할까?

****어플리케이션이 직접 하드웨어의 리소스에 접근하지 못하게 만든 것이 가상 메모리이다.

학생 개발자들에게....

***기초와 기본을 탄탄하게 닦아야 할 시기!

***개발자에게 가장 중요한 것은? **호기심

***박학다식으로 시작하여 점차 눈높이를 높여나가자!

=> 밑으로가 아니라 시야관을 넓혀라! 밑으로만 파는 것은 아집이다.

***개발은 결국 '사람'이 하는 일! -> '태도'의 중요성!

***옳고 그르다가 아니라 '다르다'의 문제다!

***욕심은 좋으나 '우선순위'를 정하자!

***고정관념을 조심하라!

***의사소통 = 말하기 + 글쓰기

***무엇보다 중요한 것은 '재미'이다!

세션 3-2 [오픈소스 & 커뮤니티와 함께 성장하기 - 권순선]

soonson@google.com

기업 문화 & 구조 비교
프로페셔널의 조건?
여러분이 할 수 있는 일

결론부터 말하자면.....

관리자 문화 : Samsung
Vs
기획자 문화 : NHN
Vs
개발자 문화 : Google

매니저

문화에 따라 분위기가 다르다!

허나, 어느 특정 문화가 좋고 나쁨이 아니라,
자본주의 사회에서 자본이 결과인데 모두 다 그 결과를 잘 이끄는 회사이다.

채용

구글 : 경력직 / 삼성, 네이버 : 신입 => 기업의 문화가 결정지음

=====> 위의 모든 부분들이 평가와 보상

***내가 어떤 회사에 가장 잘 맞을 것인가?

평가와 보상

삼성 : 조직장(임원) 결정짓는다. // 그 사람한테만 잘 보이면 된다

네이버 : 동료평가가 적지만 존재는 한다

구글 : 직속 매니저가 영향력이 있지만 동료평가가 큰 영향을 미친다
(개발자 입장에서 유리하다)

[어떤 사람이 이 회사에서 가장 좋은 임직원인가?]를 면접시 질문해 보라.
이 회사가 어떤 직원을 원하는지 어느 정도 알 수 있다

프로페셔널의 조건?

열심히, 즐겁게

Lonely Guru
Vs.
Team Player

지금은 혼자하는 때가 아니까, 과거 초창기 시절에만 그것이 가능했다.
그 시간에 팀플로 하면 시간을 아낄 수 있으며 효율이 좋다.
요즘은 과거의 그런 시대가 아니다!!!

예술가와 공학자

딜을 할 수 도 있어야한다.

ex) 3달 만들걸 1달에 만들라 하면 1달 동안 일단 만든다고 하되 2달동안
우리가 원하는 코드를 만들 수 있는 유지보수 기간을 달라...

정치/사회에도 관심을....

지금 우리 업계가 가는 길이 맞는 것인가.....

오픈 소스!

배울 수 있는게 많다.

배울 수 있는 교재 같은 것이다.

****커뮤니티!

내가 못하는 것은 그저 좌절이다.

짜증이 나서 일을 못하는 것은 사람 때문일 것이다.

사람대 사람은 커뮤니티.

온라인 상으로 새로운 사람들을 만날 수 있는 좋은 환경이다.

학생은 학교에서 배우는 것도 중요하지만 학교 밖에서 배우는 것도 중요하다.

많은 것을 배울 수 있다.

결론적으로는....

- *잘 할 수 있는 일

- *지속할 수 있는 일

- *변화를 두려워하지 않는 용기

ex) 조직은 쉽게 바뀌지 않는다. 그럴때는 이직을 할 용기가 필요하다
그렇려면 나를 알아야하고....

****당신이 원하는게 무엇인지 알고, 당신이 원하는 일을 해라!****

2014. 01. 25
전액에서~.