

به Developers Corner خوش آمدید

این راهنمای ما هست برای چگونگی پیوستن به جامعه برنامه سازان متن باز، ساختن اپلیکیشن هایی که بهتر هستند چرا که باز، شفاف، امن تر، خوب طراحی شدند و در کل با حال هستند.

منابع مطالب و ترجمه ها

این مطلب از جاهای متفاوتی گرفته شده. در ضمن بعضی از این مطالب در زبان های غیر از اینگلیسی هم هستند. اگر ترجمه نیاز به بهتر شدن داره لطفا درخواست بدید.

جدولی از مطلب

ما محتوای این مطالب را به زبانی میشه گفت که شکستیم تا راحت تر فهمیده بشه برای کسانی که در متن باز، هک مدنی (civic hacking) و برنامه سازی موبایل، جدید هستند.

- مبانی متن باز
- متن باز چی است
- راهنمایی مبتدی برای متن باز
- شروع کار با Civic Hacking
- چطور پروژه متن باز را شروع کنیم
- سه المان اصلی که همه پروژ های متن باز را تشریح می کنه
- شروع کردن استفاده از گیت هاب
- شروع کردن استفاده از Discourse بسمت کد منبع باز در اندروید بوسیله اف دروید (F-Droid)
- چیزهایی که تازه وارد ها معمولا درباره متن باز می پرسند کتاب های اصلی برای خواندن و سازمان های متن باز
- طراحی و کاربرپذیری
- مقدمه ای برای اصول طراحی آندروید
- امنیت را خسته کننده کنیم
- آیا واقعا طراحی بطور متن باز (open source) امکان پذیره ؟
- تفکر طراحی و همکاران جامعه زبان باز (open source) کجا می توانند به هم پیوندن ؟
- اعمال نفوذ طراحی کاربر محور در پروژه متن باز
- پنج قدم تا استفاده از طراحی در پروژه متن باز خودتان
- اصول اپلیکیشن
- ساختن اپلیکیشن ها برای دنیای پیچیده
- توسعه در اندروید را شروع کنیم
- امنیت و حریم خصوصی - رمزگزاری تمام بیت ها، آشنایی با امنیت موبایل
- چطور از اطلاعاتم و اپلیکیشن هام محافظت کنم (needs more content)
- چگونه از اطلاعات رمزگزاری شده محافظت کنم (مقدمه ای برای IOCipher دیسک های مجازی کدگزاری شده) چطور اطلاعات برنامه خود را حفاظت کنیم (پیش مقدمه برای sqlCipher دیتابیس رمزگزاری شده موبایل)
- چطور می تونم ترافیک اینترنت خودم رو امن کنم (مقدمه برای NetCipher)
- گرفتن عکسی حرفه ای
- مدیریت تیم و پروژه

• ساختن یک جامعه متن باز
• طریقه استفاده از IRC (Internet Relay Chat)
• داستان یک شخص و ایجاد یک جنبش نکاتی برای مریبان: داستانی از شخص
• پنج ابزار برتر برای مدیریت پروژه های متن باز
• پنج راه برای تعامل بیشتر در گروه # یک راهنمای ساده برای کسانی که می خواهند به
پروژه های متن باز کمک کنند
از Natasha Murashev

منبع [General Assembly Blog](#)

Natasha is a iOS Engineer at [Capital One Labs](#) and [instructor at GA in San Francisco](#). This post originally appeared on her blog, [Natasha The Robot](#)

When I first started as a Rails developer, I thought of all the Rails gems as magic. Some smart people were making all these cool libraries for me to use! I had no idea how any of these libraries worked, and I was ok with that. They worked and did what I needed them to do. They seemed so comprehensive and thought out, that I didn't even know what I would contribute to them even if I wanted to

To this day, I haven't contributed any open source code to the Rails community. And that's because the Rails community is extremely active on open source, so it's actually hard to find things to contribute to! Of course you can go through issues and try to solve them, but they are usually pretty complex and intimidating, to be honest. With all these smart people commenting on the issues, it's hard to feel like you're good enough to solve it

Fast forward to the end of Mobile Makers, when I first learned iOS, and I made my very first open source contribution! Since then, I've made a few more, and I'm no longer scared of contributing to even more projects! Here is how I made the jump and what I learned in the process

این یک جادو نیست

Getting out of the Rails community, one thing I learned quickly was that there is actually a sort of mistrust of open source libraries. While I wanted to use all the CocoaPods all the time (just like in Rails!), the senior iOS engineer on my team would always question using external libraries, preferring to build our own if possible

There are very few iOS libraries that are used consistently across iOS projects, and most projects only use a few libraries to start with! That's changing a bit thanks to CocoaPods and how easy it is to delete/upgrade dependencies, but it's nowhere as close to Rails level, where most of your project is made up of plug-and-play libraries

Changing my mindset to realizing that these external libraries are not magic, has made a HUGE difference for me. Knowing that the person writing a library is a real person, who makes mistakes, might not write the best-optimized code, and who can't see all the edge cases all the time makes it much easier to spot the small (or big) places where I can contribute

CARING IS SHARING

While there are some people who advocate to actively look for open source projects to contribute to, I found that all the projects I contribute to are the ones I'm actually using in my own code

The truth is, I don't have time to actively scour GitHub for issues/libraries to contribute to. But when I'm using a library in my own project, and I wish it had feature X or that bug Y was resolved, it's a no-brainer to solve it and contribute back! In fact, I get super excited when I find these opportunities, since I do like to contribute to open source

...Which brings me to my next point

...IT'S SUCH A GOOD FEELING

Contributing to open source is seriously addictive! Already, knowing how to code and bending the computer to your will is magical and makes you feel like a sorceress. But when you can take someone else's "magical" code and make it better and they agree that you made it better (by merging it in), the feeling you get is indescribable. It's like you just became a Level 10 magician instead of Level 1

START SMALL

Again, I thought of open source contributors as magicians making sweeping changes and improving everything with their magic wand, but the truth is most changes are pretty small. They just add up and eventually change and improve the whole library for everyone in a very substantial way. So don't underestimate the power of small changes

:Here are some recent examples of contributions I made/am in the process of making

MODIFY A README

I wanted to possibly add the [Toast library](#) to my iOS project, but they didn't say on their README that there was a CocoaPod available. Since I just wanted to test out the library in my project, I wanted it to be easily removable. So even though I've used this particular library before and knew it was good, I started searching [CocoaPods](#) for another Toast library

A few libraries down, I found that this particular Toast library was, in fact, a CocoaPod! To make sure other iOS developers know there is a CocoaPod available, I submitted a pull request with CocoaPod installation instructions to the README file of the library. Tiny change, but hopefully, it'll help other developers

ADD ADDITIONAL SAME FUNCTIONALITY

When building an Android app for the [CodePath](#) final project, my team wanted to try out the current official Android font – [Roboto](#). Turns out it's pretty manual to incorporate external fonts into Android, so we used a library called [RobotoViews](#) to help us out. Basically, each view has to be configured to have the Roboto typeface available

However, there was one view that the RobotoViews library did not include that we needed – the

newer Switch view. Adding the Switch view was just a lot of copy/paste following the convention of the other views, so it wasn't hard to add, but now another view is available as a
!RobotoView

In other words, the creator of RobotoViews already did all the hard work to make it easy to
.modify only a few things to add a new view

Similarly, I contributed to a popular [iOS Foursquare client library](#) by just adding an additional function that wasn't there originally, but was very easy to add based on the creator's work to
!abstract this process

REFACTOR

I was adding a very small change to the [ECSlidingViewController](#) to make sure a keyboard is dismissed when a sliding menu slides out, when I noticed that three functions had the same exact code with a very slight variation. So I just refactored that code by creating a function those three functions can just call passing in the one different argument, so anyone who needs to make a
.change to this function in the future, only needs to make the change once

As you can see, all my open source contributions are very tiny and easy to make! As you keep using external libraries, you'll see similar opportunities. So go ahead and make the tiny changes
!– they count

HOW TO CONTRIBUTE

Watch this great [step-by-step RailsCast on how to contribute to open source](#) (it's a very similar
:process for non-rails projects). But basically, here are the steps

FORK

Find the library you want to contribute to on Github, and simply click the Fork button! github-
fork

CLONE

Next, clone the library that you forked – it should now be under your name (e.g. NatashaTheRobot/ECSlidingViewController), not the original creator! git-clone

BRANCH, CHANGE, PUSH

Once you've cloned the repository, change into the repository's folder. Next, check out a new branch with a good name reflecting the change you will make. Make the change, and push the
.branch to GitHub

When you go to your GitHub profile main page, you'll see a big green "Compare and Pull Request" button. Take a look at your files, make sure everything looks good. Then make sure you make a pull request to the original branch (not the one you forked). When you've done the pull request, it should look something like this! Again, make sure you're doing a pull to the

original creator's master branch! git-pull-request

TWEET

This step is of course optional, but I like to tweet at the creator to notify them of the change. It's possible they don't have GitHub notifications turned on, especially if it's an older repository with less recent changes, so it's nice to let them know and start a conversation

They might also be too busy to merge it in, so it's nice when they reply and tell you they'll take a look when they have a chance. You know that your pull request will not be in limbo forever

!ENJOY

Again, contributing to open source is really fun and a great way to learn from the best and #improve your own skill. Hope this guide makes you feel less intimidated at contributing
شروع کار با Civic Hacking

منبع : [OpenSource.com](https://opensource.com)

civic hacking چیست؟

هفتاد تن از مردم در یک روز آفتابی در اکالند با هم جمع شدند تا داوطلبانه شهر را بهتر بکنند. اونجا از چنگک یا ابزار باغچه خبری نبود. نه برس رنگی، نه سطل زباله ای، نه کیسه ی پر از قوطی. این هفتاد نفر با کلی لپ تاپ بودند تا بصورت داوطلبانه وب سایت شهرشون را بهتر کنند.

تعامل این گروه با هم وب سایت (پاسخ های اکالند) را ساختند. (<http://answers.oaklandnet.com>), راهی بسیار ساده برای پاسخ گویی برای سوالات متدوال در وب سایت شهر اکالند. از نحوه پرداخت تیکت های پارکینگ گرفته تا کارهایی در شهر، وب سایت جدید به شهروندان توجه داره و توسط جامعه ساخته شده.

این برنامه یک روزه "writeathon" نام داشت و اکثریت کسانی که شرکت داشتند برنامه نویس وب نبودند ولی کسانی بودند که آمده بودند به سوالات شهروندان اکالند پاسخ بدند. قطعا تکنولوژی هم اونجا بود، راه اندازی یک سرور و انشعاب کدهای متن باز به وب سایت. این برنامه نویسان وب، پاسخ دهندگان و کارکنان شهر هر کدام بخشی از رشد یک حرکت civic hacking بودند.

Civic hacking کسانی هستند که باهم سریع و خلاقانه برای بهبود دولت کار می کنند.—
[Jake Levitas](#)

در اینجا می توانید چند پروژه معروف متن باز civic hacking را مشاهده کنید:

- [OneBusAway](#) (on [GitHub](#))
- [visualizations](#) (on [GitHub](#))
- [Service Tracker](#) (on [GitHub](#))
- [311](#)
- [Flu Shot Finder](#) (on [GitHub](#))
- [StreetMix](#) (on [GitHub](#))
- [CityVoice](#) (on [GitHub](#))

متن باز در civic hacking

برنامه های متن باز اساساً سیویک هکنینگ هستند. داوطلبان با انگیزه 'کدهایی را می نویسند و چیزی را اختراع می کنند که محله خودشان را بهبود کند. ولی اینکار را طوری انجام می دهند که برای جوامع دیگر مفید باشد. این امکان را میدهد که کدها را با هر جای دیگر به اشتراک بزارند بدون هیچ م

به عنوان نمونه 'چند سال پیش بوستون شاهد یک ریزش شدید برف بود تا حدی که شیرهای آتش نشانی هم پوشیده شده بودند. همون برف باعث قطع شدن خطوط برق و آتش سوزی شد. یک سری از هکرها مدنی (civic hackers) این مساله را دیدند و این را درست کردند. [Adopt-a-hydrant](https://github.com/codeforamerica/adopt-a-hydrant) در [GitHub] (<https://github.com/codeforamerica/adopt-a-hydrant>) یک راحل برای اینکه هماسیه ها شیرهای آتش نشانی در محل زندگیشان را از برف پاک کنند. در تابستان بعدش همون کدها به هونولولو منتقل شد 'این بار نه برای برف بلکه برای طوفان و سیل. (<http://sirens.honolulu.gov>). از اون موقع Adopta بارها به جایی دیگری در کشور منتقل شده و با بهبود کدها دوباره از آن استفاده شده.

شروع کنیم

پروژه ای خوب برای اضافه کردن یک سرویس به شهرتون. [Click that 'hood](#) (on [GitHub](#)) به عنوان نمونه این یک بازی خوب برای دونستن در مورد جایی که درش زندگی می کنید. خوب بودنش اینکه یک راهنمای خیلی خوب داره (https://github.com/codeforamerica/click_that_hood/wiki/How-to-add-a-city-to-Click-That-Hood) برای اضافه کردن شهر خودتون به این بازی. این راهنما شامل نحوه استفاده از ابزار زبان باز 'همکاری در GitHub و پیدا کردن اطلاعات آزاد و همه مهارت ها برای شروع کا در سیویک هکنینگ هست.

پیدا کردن اطلاعات باز

جنبش civic hacking وابستگی به این داره که امکان آسان برای پیدا کردن اطلاعات درباره دولت ها و مکان های حکومت می کنند وجود داشته باشد. به عنوان مثال اگر دیتا های حمل و نقل در دسترس نباشند ما نمی توانیم اپلیکیشنی برای زمانبندی حرکت قطارها و اتوبوس ها بسازیم. خوشبختانه خیلی از شهرها به این نتیجه رسیدند که اطلاعات مهم را برای استفاده باز بگذارند. Data.gov لیست بسیاری از پورتال های دولت ها ر در کشورهای متفاوت دارند. این پورتال همه اطلاعات در دسترس در مورد شهر را در یک جا قرار داده. بهترین دیتا پورتال ها طوری هستند آماده خواندن و استفاده در اپلیکیشن ها هستند. وب سایت های شهرت را چک کن و ببین اگر این چنین پورتالی وجود داره یا نه. اگر نه 'با شهرت کار کن برای راه اندازیش که بتونه با پروژه های خوب سیویک هکنینگ کار کنه.

پورتال اطلاعات متن باز:

[CKAN](#)•

[DKAN](#)•

[Socrata](#)•

جامعه

این خیلی مهمه به یاد داشته باشی که سیویک هکنینگ شامل جامعه و تکنولوژی میشه. تمام سیویک هکرها ی نخبه باید باهم باشند تا بتوند مشکلات جامعه ما را رفع کنند. این کد را برای

آمریکا چک کنید برای پیدا کردن داوطلبان محلی تا بتوانید گروه خودتون را شروع کنید [Brigade](#) بهترین مساله برای یکی شدن با هکرهاى ديگه اين هست كه مى تونيد بفهميد كه اونها در چه چيزى به موفقيت رسيدن كه شما هم مى تونيد در شهرتون انجام بديد. Brigade يكي از بهترين منابع هست براي پيدا كردن ابزارهاي متن باز و پروژه اي كه بشه درش كار كرد. همچنين كار كردن با كاركنان شهر و مسولان سيويك كه بخشي از Brigade هستند حيايه چرا كه ساخت اپليكيشن هاي تكنولوجي سيويك توسط هكرهاى سيويك مشكلات اجتماعي را حل مى كنه. توانايي هاي متفاوت و ديدگاه هاي مختلف براي حل مشكلاتي كه بر مردم اثير مى زاره 'مورد' نياز هست. و در آخر 'براي هيچان زده شدن در مورد جنبش سيويك هك در آمريكا' (<http://www.codeforamerica.org/about/fellowship>). هك كردن به خوشحالي! # شروع كار با Discourse

منبع: [Discourse wiki](#)

اين صفحه به نكات ابتدای استفاده از Discourse می پردازد. به اين توجه كنين كه ممكن رابط کاربري يا interface با screenshot ها فرق بكنه 'اين دليلش اينه هست كه ممكن مدير سايتت يك سري تنظيمات را به دلخواه انجام كرده.








ليست موضوعات

چند ليست از موضوعات در Discourse. اولين چيزي كه عنوان كاربر جديد خواهي ديد Top ليست خواهد بود. اين بهترين پست ها از قبل نشون مى ده كه بر اساس تعداد لايك ها و بازديد كنندگان است.

es

Top






+ Create Topic

Users	Posts	Likes	Views	Activity	
    	113	264 	1.3K	23d	4d
    	158	128 	2.8K	23 Jan	4d
    	90	153 	6.0K	Jul '13	2d
    	118	104 	5.1K	Jun '13	11d
    	106	106 	1.2K	Jun '13	Jun '13
    	79	99 	620	15d	2d

در این صفحه چند چیز هست. در ردیف اول **Category filter** را داریم ' Nav menu items و دکمه **Create Topic**

شما می تونی از **Category filter** برای فیلترکردن مقوله ها استفاده کنی. به عنوان مثل' این بالای صفحه است که فیلتر شده فقط موضوعات مربوط به UX را نشون بده.

+ Create Topic

	Posts	Likes	Views	Activity	
	106	106 ❤️	1.2K	Jun '13	Jun '13
	79	99 ❤️	620	15d	2d
	80	78 ❤️	1.4K	Oct '13	3d
	52	83 ❤️	353	28 Jan	14d
	49	42 ❤️	832	Jun '13	Jun '13

موضوعات در یک ردیف نشون داده می شه. از چپ به راست 'عنوان' را داریم 'موضوعات' ** مقوله ها ' لیست شرکت کنندگان ** آواتار ' تعداد پست ها در موضوعات ' تعداد لایک ها در موضوعات ' تعداد موضوعات نمایش داده شده ' و تاریخ ** ایجاد شدن ** و آخرین فعالیت .

با کلیک کردن بر موضوع یا شمارنده پست به شما آخرین پستی را که خواندید در موضوعات نشان داده می شه. کلیک کردن بر شمارنده لایک ها شما را به ابتدای موضوع در حالت خلاصه می بره ' [summary mode](#) اگر موجود باشه. با کلیک کردن بر تاریخ شروع موضوع این شما به را به ابتدای شروع موضوع می بره ' و با کلیک کردن بر آخرین تاریخ فعالیت ها این شما را به پایان اون موضوع می بره.

لیست های موجود دیگر شامل آخرین (این بطور پیش فرض خواهد بود وقتی که شما به اطمینان درجه 1 دست پیدا کردید) ' [جدید و خوانده نشده](#)

جدید و تایپیک های خوانده نشده

موضوعات جدید اونهایی هست که هیچ وقت نخوندیشون. بطور پیش فرض ' فقط موضوعاتی که طی دو روز گذشته ساخته شده جدید محسوب می شه.

موضوعاتی که جدید محسوب می شن این شکلی هستند : (screenshot)

دو نوع موضوعات خوانده نشده وجود دارد: اونهایی که خواندنش را متوقف کردی و اونهایی که تازه به اشتراک گذاشته شدن.

موضوعاتی که خواندنش را متوقف کردی یک حباب خاکستری دارد که داخلش شماره هست. این نشان دهنده تعداد پست هایی هست که هنوز نخواندی. این مثل این می مونه: [یک حباب خاکستری در پایان عنوان موضوع](#) تاپیک هایی که پست جدید دارند از آخرین باری که خونیشون یک حباب دارند با یک شماره داخلشون. این تعداد پست های جدید را نشون می ده. این مثل این می مونه: [یک حباب آبی در پایان عنوان موضوع](#)

ناو منو

ناو منو در بالا نزدیک لیست عنوان نمایش داده می شه :

تب های موجود آخرین ها هستند. جدید خوانده نشده ستاره دار طبقه بندی ها بالا خوانده و پست شده. تمام این نمایش داده نمی شن ولی قابل دسترسی هستند.

آخرین ها: تمام موضوعات بر اساس آخرین فعالیت است.

جدید: اینجا فقط موضوعاتی در نظر گرفته می شوند که [جدید](#) در لیست قرار می گیرند. (فقط به سیستم وارد شده)

خوانده نشده: اینجا فقط موضوعاتی در نظر گرفته شدند که [خوانده نشده](#) در این لیست قرار می گیرند (فقط به سیستم وارد شده)

ستاره دار: فقط موضوعاتی که ستاره دارشون کردی در لیست قرار می گیرند. (فقط به سیستم وارد شده)

خوانده شده: موضوعات بر اساس آخرین باری که خوانده شدن منظم می شن. (فقط به سیستم وارد شده)

دسته بندی ها: نمایش کلی دسته بندی ها در تالار گفتگو.

بالا ترین: براساس آخرین فعالیت در روز گذشته هفته ماه و سال نشان داده می شوند.

در قسمت سمت چپ ناو منو **زبانه پیش فرض** هست و این چیزیه که وقتی به روی لوگوی Discourse و یا نشان طبقه بندی کلیک کنید می بینید. بصورت پیش فرض این آخرین زبانه هست.

مقوله ها

هر موضوعی یک **طبقه بندی** دارد. Discourse 'برخلاف تالارهای گفتگوی قدیمی' طبقه بندی مسطح را ترویج می کنه : فقط یک لوول زیر شاخه موجوده و تمام موضوعات در همون لیست قرار می گیرند.

صفحه مقوله ها

صفحه مقوله ها بهت یک نمای کلی از تمام مقوله ها در تالارهای گفتگو نشون می ده. (screenshot)

اینجا ما فقط یک ورودی برای صفحه طبقه بندی های می بینیم که ** نشانه طبقه بندی ** هست 'تعریف' این نشانه های **زیرشاخه هاست** و **شرکت کنندگان** آواتار تکراره گزیده های اخیر موضوعات و ریت پست های جدید و موضوعات.

صفحه موضوعات

ساخت یک موضوع

برای بوجود آوردن یک موضوع جدید، به روی + Create Topic در بالای هر لیست موضوعی می توانید کلیک کنید. اگر لیست موضوع بر اساس طبقه بندی فیلتر شده، اون طبقه بندی از قبل برای شما پر شده.

یک موضوع خوب انتخاب کن و شروع کن به نوشتن آهنگسازی. # شروع کار با GitHub

منبع : [LifeHacker.com](https://lifehacker.com)

Lifehacker عزیز، من کد نویسی را یاد گرفتم و می خواهم با GitHub برای مدیریت پروژه هام استفاده کنم. با وجود آموزش مقدماتی که ارائه دادن هنوز نمی دونم چطور کار می کنه. می تونی بهم کمک کنی ؟

با احترام، کمک Git

GH عزیز

GitHub's ابزار خویه ولی قطعا یک مقدار گیج کننده است برای بار اول (و شاید چند بار بعد از اون). بخاطر همین GitHub نرم افزاری را برای (OS X و Windows) تا کار کردن با اون ساده تر بشه. با وجود این، راحت تره که به روش قدیمی یادش بگیریم در غیر این صورت گزین های ساده این نرم افزار بی معنی می شه برای شما. بزارید با هم کار کردن با نکات ساده اش شروع کنیم

قدم اول: برای GitHub ثبت نام کنید

اینجاست که بخش آسانش میاد: برای خودت یک حساب با GitHub درست کن [از بالای صفحه ثبت نام کن](#). بعد از تمام کردن ثبت نام، GitHub بهتون اجازه ورود می ده و شما را به یک news feed خالی می بره. در میان صفحه شما یک boot camp می بینید (عکس در راست). ما به بهت نشان می دهیم که چطور حساب کاربریت را تنظیم کنی و دیرتر انبارت را بسازی. به روی "Set Up Git" کلیک کن تا شروع بشه.

قدم دوم: Git را نصب کن

بودن GitHub بخاطر وجود git یک کنترل کنترل کننده آپ هست. سایت تقریباً بر اساس git کار می کنه، و git تقریباً قدیمی محسوب می شه. اون با command line کار می کنه و هیچ ظاهر گرافیکی باحالی برای رابط کاربری یا user interface نداره. از وقتی این برای مدیریت کدهایی که می نویسی ساخته شده، پس نباید زیاد ترسناک باشه. البته، همطور که قبلاً گفته شد، GitHub یک نرم افزار بسیار عالی درست کرده که به شما اجازه می ده از سرویسش استفاده کنی بدون نیاز به command line ولی این زیاد بهتوت کمک نمی کنه قبل از اینکه نکات ابتدایی را بدونید.

Git با خواندن کدهای محلی repository (فقط پوشه ای که کد های پروژه در آن است) در کامپیوترت و بازتاب کردن کدهات به جاهای دیگه (در این مورد GitHub's servers) کار می کنه. در ابتدا ما کل repository محلیت را به GitHub ها (یا ارسال) می کنیم ولی این فقط

یکبار اتفاق می افتد. همزمان با اینکه شروع به کار می کنی تو تغییرات را commit می کنی. بعد از اون GitHub تمام تغییرات که ایجاد کردی دنبال می کنه' با ساختن ورژن های متفاوتی از فایل ها بهت کمک می کنه به ورژن قبلی برگردی (یا فقط تغییرات را دنبال می کنه به دلایل دیگر). این اولین دلیلی هست که چرا تو می خوای ورژنی از control system مثل git استفاده کنی' ولی دلیل دیگر استفاده از اون برای وقتی هست که می خوای کدها را با اشخاص دیگری که در پروژه ات کار می کنند مدیریت کنی. وقتی که توسعه دهندگان متعددی (developers) کدها با استفاده از git ارسال (commit) می کنند' Github به یک مرکز repository تبدیل می شه که تمام کدهایی که کسان دیگری دارند روش کار می کنند با هم sync (همگام) می شوند. وقتی تغییرات را commit می کنی و دیگران هم اون را وارد می کنند (همه با هم به repository محلی synv می شنند). و تو هم همین کار را با کدهای اونجا انجام می دی.

Git همه اینها را عملی کرد' پس تو نیاز داری که آخرین نسخه را دریافت کنی (-git/http://scm.com/downloads) و نصبش کن. برای OS X, فقط نیاز به نصب command line app را داری. و در Windows چند مورد دیگر را می گیری. در مورد نحوه کارکردشون بیشتر توضیح می دیم در قدم بعدی.

قدم سوم: Git را تنظیم کن

برای نصب git ' شما نیاز دارید که مسیرون را به command line پیدا کنید. در OS X ' این به معنای باز کردن Terminal app هست (Terminal -> Utilities -> Applications -> Hard Drive) و در Windows یعنی باز کردن Git Bash app که نصبش کردین - نه Windows command prompt. وقتی آماده هستید اسمتون را به git مثل این بگید:

```
"git config --global user.name "Your Name Here"
```

به عنوان نمونه' مال من این شکلیه چرا که من از یک حساب آزمایشی برای نمونه استفاده می کنم:

```
"git config --global user.name "Adam Dachis"
```

می تونی هر اسمی را که دوست داری را وارد کنی' ولی بعدا نیاز به وارد کردن ایمیل هست و اون باید همون ایمیلی باشه که برای وارد شدن به GitHub استفاده می کنی:

```
"git config --global user.email "your_email@youremail.com"
```

اگر به هر دلیلی برای ساختن حساب با GitHub از ایمیلی اشتباه استفاده کردی باید اون را عوض کنی. (<https://help.github.com/articles/how-do-i-change-my-primary-email-address>).

الان' برای جلوگیری از وارد کردن مجوز ورود و ساختن رمز SSH ' شما می تونید credential helper را نصب کنید که رمزهاتون ذخیره می شه. اگر از Windows استفاده می کنی' [download](#) این را نصب کن. و اگر از OS X استفاده می کنی' باید این را از طریق Terminal وارد کنی. برای شروع از این دستور برای دانلود credential helper استفاده کن:

```
\ curl -s -O
```

<http://github-media-downloads.s3.amazonaws.com/osx/git-credential-osxkeychain>
این یک فایل خیلی کوچک را دانلود می کنه که زمان زیادی را نمی بره. وقتی که تمام شده' دستورات بعدی را دنبال کن تا مطمئن بشی که مجوزت درسته برای فایلی که دانلود کرد (و درستشون بکنی اگر درست نیست)

```
chmod u+x git-credential-osxkeychain
```

حالا وقت نصب credential helper داخل همون پوشه ای که git را نصب کردی. برای نصبش این فرمان را وارد کن.

```
`\sudo mv git-credential-osxkeychain `dirname `which git
```

تو وادارارش می کنی برای رمز مدیریت یا administrator password چرا که فرمان پایین با sudo

شروع میشه. Sudo کوتاه شده ی "super user do" هست و وقتی لازم میشه که برای انجام یک کار نیاز دسترسی به root باشه. فرمان sudo به تو اجازه می ده برای مدتی کوتاه کاربر root بشی (کاربری که تقریباً هر کاری را می تونه انجام بده) در سیستم عاملت که هر کاری که نیاز داری بتونی انجام بدی. از تو رمز عبور را می خواد تا بتونه تایید که که مدیریت کامپیوترت اینچنین اجازه ای را به تو بده. وقتی که رمز عبورت را وارد کردی کمک کننده گواهی جا به جا شد' نصب را با این فرمان تمام کن :

```
git config --global credential.helper osxkeychain
```

الان که همه چیز را تنظیم کردی می تونی شروع به استفاده از git و GitHub کنی!

قدم چهارم: ساخت اولین انبارت

بالاخره الان تونستی انجامش بدی' تو الان می تونی از GitHub استفاده کنی! برای عنوان اولین کار' ما یک فضا یا repository یا "repo" درست می کنیم. برگردیم به GitHub به روی کلید "New Repository" در بالای صفحه حساب کاربریت کلیک کن. (نکته: اگر هنوز قسمت GitHub bootcamp را نشان می ده' در زیر آن نشان داده میشه)

وقتی که داری اون فضا یا repository را درست می کنی چند گزینه داری که باید در موردش تصمیم بگیری' از جمله اسم و اینکه این برای عموم قابل دسترسی هست یا نه. انتخاب اسم باید آسان باشه چرا که قبلاً اسم را برای پروژه ات انتخاب کردی. اگه فقط دارین تمرین می کنید از این استفاده کنین "Hello-World". چرا "Hello-World" نه "Hello World"؟ چرا که فاصله و کارکترای خاص ممکنه مشکل ایجاد کنن. ساده نگهش دار که بتوانی راحت در خط فرمان بنویسیش. اگر می خوای اسم سختتری انتخاب کنی می تونی در بخش توضیحات دلخواهی در پایین می توانی اضافه کنی بهش.

اگر داری یک پروژه متن باز درست می کنی' یک فضای عمومی می خوای. اگر می خوای خودتون کد نویسی کنین و یا اینکه فقط با کسان دیگری به اشتراک بزارید' فضای خصوصی این کار را انجام می ده. انتخابی را انجام بده که بهترین را برای پروژه ات به ارمغان میاره.

وقتی تمام شدی' می توانی روی دکمه "Create repository" کلیک کنی ولی شاید می خواهی یک کار دیگه را هم اول انجام بدی : چک کردن اولیه این repository (فضا) با "README" باشه. چرا ؟ تمام repositories به فایل README نیاز دارند. بصورت ایده آل این فایل حامل اطلاعات کمی در مورد پروژه شماست ولی شاید الان نمی خواهی باهاش مواجه بشی. با چک کردن اولیه repository با README ' به شما این README خالی می د می توانی بعداً بهش رسیدگی کنی. برای یکی از هدف های این آموزش ما این قسمت را unchecked باقی می زاریم چرا که در بخش بعدی ما از ابتدا یک فایل README برای تمرین ارسالتش (committing) به GitHub درست می کنیم.

قدم پنجم: اولین ارسال یا Commit را بساز

وقتی که فایل هات را به GitHub می فرستی' اونها را commit می کنی. برای تمرین repository محلیت را initialize می کنیم و فایل README را commit می کنیم. قبل از اینکه شروع کنی' باید بدونی که کد های محلی repository در کامپیوترت کجا هستند و چطور به آنها از طریق خط فرمان یا command line دسترسی پیدا کنی. در این درس ما فرض می کنیم که یک directory وجود داره به اسم "Hello-World" و در پوشه ی اصلی کامپیوترت هست. اگر نیاز داری یکی بسازی' این فرمان را اجرا کن (برای Git Bash در Windows و OS X's terminal یکی هست) :

```
mkdir ~/Hello-World
```

الان directory را عوض کن با استفاده از فرمان cd یا (change directory) :

```
cd ~/Hello-World
```

اگر نمی دوانستین ~ این چیه ' ~ این نشان دهنده home directory در Git Bash و Terminal است. این خیلی کوتاهه و نیازی تمام اون را تایپ کنی (که تقریباً این شکلی هست اگر بخوای تایپ کنی (/Users/yourusername/). حالا که repository آماده هست این را تایپ کن :

```
git init
```

اگر از قبل repository (فضا) آماده هست ' فقط نیاز داری که دستور cd در directory را انجام بدی و بعد دستور git init شروع به کار کنه. در هر صورت repository محلی تو آماده است و تو می توانی code committing را شروع کنی. ولی صبر کن ' تو هیچ چیز برای commit کردن نداری! این فرمان را برای ساختن یک فایل README اجرا کن :

```
touch README
```

بزار یکم استراحت کنیم و ببینیم دقیقاً چی شده. برو به پوشه اصلی کامپیوترت و پوشه ی Hello-World (یا هر پوشه ای که برای local repository استفاده می کنی) . متوجه می شی که فایل README اونجاست ' از فرمانی که اجراش کردی باید تشکر کرد. چیزی را که نخواهی دید git folder هست اونم دلیلش اینکه اون پنهانه. Git اینجا پنهان شده ' ولی بخاطر اینکه فرمان git init را اجرا کردی می دونی که اونجاست. اگر شک داری فرمان Git Bash/Terminal را اجرا کن تا لیست همه چیزهایی که در directory را داری ببینی.

خوب git از کجا می دونه که می خوام این فایل README را commit کنیم؟ نمیدونه ' و تو باید بهش بگی. این فرمان را برای این ترفند استفاده کن :

```
git add README
```

اگر می خوای فایل دیگری به commit اضافه کنی ' باید از همون فرمان استفاده کنی ولی جای README اسم اون فایلیت را بزاری. الان این فرمان را اجر کن برای commit (ارسالش).

```
git commit -m 'first commit'
```

وقتی دستورات دیگر کمی ساده بودند ' فرمان commit یکم مشکلتره پس یکم استراحت کنیم. وقتی که git را تایپ می کنی یعنی به خط فرمان می گی می خوای از برنامه git استفاده کنی. وقتی تایپ می کنی commit ' یعنی داری میگی می خوای از فرمان commit استفاده کنی. هر چیزی که این دو تا دنبال کنن به عنوان گزینه های دیگر حساب می شنند. اولین چیز m- هست که به عنوان پرچم یا flag شناخته می شه. flag مشخص می کنه که تو می خوای یک کار خاص انجام بدی بجای فرمان commit. در این مورد m- flag معنی یک پیام یا "message" را داره و چیزی را که دنبال می کنه پیام commit هست (به عنوان نمونه ' commit اول). این پیام به خودی خود مورد نیاز نیست (معمولاً نیاز به ارائه یکی از اونهاست) ' ولی این بطور ساده یک رفرنس هست که بهت کمک می کنه برای تفکیک یک سری فایل (یا فایل های) متفاوتی که commit کنی در repository.

اولین commit (ارسال) تو باید خیلی سریع بره چرا که تو هیچ چیزی آپلود نکردی. براب فرستادن README file خالی به GitHub ' باید چند تا دستور بدی. این اولینش هست :

```
git remote add origin https://github.com/yourusername/Hello-World.git
```

تو نیاز به جایگزینی "yourusername" داری — اونی که حدس زدی — حساب کاربریت در GitHub. برای من ' این شکلیه باید باشه:

```
git remote add origin https://github.com/gittest1040/Hello-World.git
```

این دستور به git می گه که Hello-World repository را کجا بفرسته. الان تنها کاری نیاز به انجام داری ' ارسال این هست :

```
git push origin master
```

وقتی که این فرمان را اجرا کردی ' همه چیز (در این مورد فقط فایل README) به GitHub ارسال می شه. تبریک برای اولین commit!

بیشتر بیاموزیم

GitHub به بیشتر از commit کردن README file نیاز داره' ولی این مسائل ابتدایی بهت یاد می ده که چطور با اپ git کار می کنه. الان می دونی که GitHub چطور کار می کنه و این مهمترین' اگر ترجیح می دی الان می تونی کدهات را با GitHub مدیریت کنی. اگر می خوای بیشتر در مورد GitHub یاد بگیری' یک سری آموزش های خوب هست. برای شروع یک نگاهی به اینها بینداز. [LockerGnome's GitHub guide](https://help.github.com/articles/fork-a-repo) (https://help.github.com/articles/fork-a-repo) and خوش باشی با مدیریت کدهات!

عشق ' Lifehacker

منابع اضافی از GitHub

منابع خوب برای یاد گرفتن git و github (https://help.github.com/articles/good-resources-for-learning-git-and-github/) بسمت کد منبع باز در اندروید بوسیله اف دروید (F-Droid) منبع <https://opensource.com/life/15/1/going-open-source-android-f-droid>

اندروید نامی که میتواند در دنیای آزاد و کد منبع باز تفرقه افکن باشد. بعضی روی سر میگذارندش و بعضی ها از کنارش هم رد نمیشوند. بعضی ها هنوز از نسخه های باز و آزاد اندروید مثل Cyanogenmod (<http://www.cyanogenmod.org/>) یا Replicant (<http://replicant.us/>) استفاده میکنند. اگر از دستگاهی با سیستم عامل اندروید استفاده میکنید- فرقی نمیکند که چه نسخه ای- یک فاکتور از همه چیز بیشتر بشما کمک میکند که بیشترین بهره رو از دستگاه خود ببرید: برنامه ها (آپس) و بس.

بیشتر کاربران برنامه های مورد نظر خودشون رو از بازار گوگل یا Google Play دریافت میکنند. بعضیها برنامه های خودشون رو از بازار Amazon یا منابع متفرقه دریافت میکنند. اکثر این برنامه ها که از بازار Google یا Amazon گرفته میشوند اختصاصی هستند و اقلب اطلاعات زیادی در مورد شما را جمع آوری میکنند.

پس چه گزینه ای باقی میماند اگر بخواهید از اندروید استفاده کنید و در عین حال برنامه های باز و رایگان داشته باشید؟ از F-Droid استفاده کنید (<https://f-droid.org/>) F-Droid چیست؟

F-Droid؟

شاخه ای نصب شدنی از نرم افزارهای FOSS (نرم افزارهای کد رایگان و منبع باز) بر روی پلت فرم های اندروید میباشد.

تمام برنامه ها فقط فقط "کد منبع باز و رایگان" هستند. کد منبع در دسترس است و معمولاً لیست برنامه ها حاوی لینک دانلود هستند. اگر برنامه ای وابسته به سرویس و یا دسترسی خصوصی باشد اف دروید به شما اخطار میدهد.

پیام اخطار باز و رایگان نبودن

شروع

دو راه برای شروع نصب برنامه ها هست: دانلود و نصب برنامه اف دروید مستقیم از (<https://f-droid.org/repository/browse>) و یا دانلود و نصب فایل apk. و نصب بعد از دانلود. در هر صورت شما احتمالاً احتیاج خواهید داشت که گزینه اجازه نصب برنامه های متفرقه را در گوشی خود فعال کنید. برای اینکار به قسمت تنظیمات (settings) رفته، سپس به قسمت امنیت (Security) و در آخر گزینه منابع نامعلوم (unknown sources) را فعال کنید.

من اف دروید رو بخاطر آسان بودن جستجو و بروز کردن برنامه ها ترجیح میدم. استفاده از اف دروید

فرض کنیم که شما قصد دارید که از اف دروید برای نصب برنامه ها استفاده کنید. بعد از نصب نرم افزار اف دروید آن را باز (اجرا) کنید. احتمالاً چند دقیقه برای برنامه طول خواهد کشید که

اطلاعات خود را بروز کرده تا چک کند که برنامه ها بانک اطلاعاتی با گوشی شما همخوانی و مطابقت دارد یا نه

بعد از انجام این مرحله شما آماده استفاده از اف دروید خواهید بود.

برنامه ها، برنامه ها، برنامه ها

اولین نکته ای که توجهتون رو جلب خواهد کرد تعداد برنامه های موجود در اف دروید است:

فقط بیش از 1300 برنامه، در برابر هزاران برنامه موجود در Google Play Store. شما احتمالا اکثر برنامه هایی رو که دوست داشتید و بهشون عادت کرده بودید رو در اف دروید پیدا نخواهید کرد، اما این الصاما به این معنی نیست که بعضیهاشون رو پیدا نکنید. از جمله مرورگر

فایرفاکس (Firefox)، برنامه ذخیره کننده اون کلاد (Owncloud)، برنامه وی ال سی نمایش دهنده فایل های تصویری (VLC)، جستجوگر داک داک گو (DuckDuckGo)، برنامه دریافت ایمیل کی ناین (K-9 Mail)، و برنامه اف بی ریدر برای خواندن کتب الکتونیکی (FBReader).

برنامه ها به چندین دسته مختلف تقسیم شدند، از آموزشی تا بازی، از اینترنتی تا برنامه های کاربردی و کاری. روی مثلی که بسمت راست در گوشی پنجره اف دروید بزنید تا لیست طبقه بندی ها برای شما باز شود.

طبقه بندی های اف دروید

از آنجا روی طبقه بندی دلخواه بزنید و سپس روی برنامه ای که می خواهید نصب کنید. اف

دروید قسمت قابل نصب رو دانلود میکند. سپس لیستی از درخواست دسترسیهای را که برنامه تقاضا میکند بشما نمایش میدهد، و آیا شما میخواهید ادامه بدهید یا نه. هروقت که حاضر بودید روی گزینه نصب یا (Install) بزنید.

نگهداری و نکات جانبی

پنجره اصلی برنامه اف دروید حاوی لیستی از برنامه های قابل دسترسی، برنامه هایی که نصب کردید، و برنامه های نصب شده ای نیاز به بروز شدن دارند. شما میتونید برنامه هایی که احتیاج ندارید را حذف کنید یا انهایی که احتیاج دارید را با یک (یا دو) حرکت بروز کنید. شما حتی میتونید لیست برنامه های در دسترس را بروز کنید. برای اینکار روی تجدید یا (refresh) بزنید تا لیست برنامه ها بروز شوند.

حالا چرا این کارو میخواهید انجام بدید، غیر از اینکه کرمتون گرفته. درحالی که برنامه های جدید اینقدر زود بزود به لیست اف دروید اضافه نمی شوند، شمار برنامه های در دسترس در اف دروید فقط در یکی دو ماه اخیر از حدود 1200 به 1340 عدد پرش کرده.

اگر چه تنوع و تعداد نرم افزار های قابل دسترس در اف دروید به Google Play Store و بازارهای متفرقه دیگر نمیرسه ولی اگر میخواهید از نرم افزار های رایگان و کد منبع باز در دستگاه اندروید خود استفاده کنید اونوقت اف دروید یک گزینه قابل تامل است.

چطوری در (در 2014) پروژه های متن باز را شروع کنیم

منبع: OpenSource.com

For open source projects to survive and thrive, it takes all sorts of different people contributing in various ways. We hope you are already participating in the open source community, but if you're not, 2014 is going to be a great year to start

?Already an open source convert? Why not consider giving back in a new way this year

.Here are seven roles you might consider taking on in the new year

7 راهی که شما می توانید درگیر پروژه های متن باز در 2014 بشوید

1. **Get your start as an open source user.** First and foremost, we hope you're an open source user! There are open source solutions for nearly every need you can imagine, and chances are you've already used a few. From browsers like [Firefox](#) to email clients like [Thunderbird](#), and content management systems like [Drupal](#) and [Wordpress](#), much of the software that powers the Internet is open source. Why not give a new tool a try this year

Share your favorite project with the world as an open source promoter. The open source .1 community and every project within it needs cheerleaders to spread the word about the benefits of the open source way. Are you a social media junkie? Join us on [Facebook](#), [Twitter](#), or your favorite social media platform, and share your own open source story with the world

Put your digital quill to work as an open source writer. Writing for open source projects .1 can take many different forms, but two roles that are almost always in need are the documentation writer and the translator. There are lots of tools available to help you document open source projects. A couple we've profiled this year are: [Zanata](#) for translators and [AsciiDoc](#) .for documentation

For more general-purpose writing tools, [LibreOffice](#) is a complete office suite that can do word !processing and so much more

Let your artistic skill shine as an open source designer. Do you have a knack for design? .1 A strong interface and visual cues can [make or break an open source project](#), and many projects could benefit from your help. If you've got the skills but not the tools, check out [GIMP](#) (the GNU Image Manipulation Program) or [Inkscape](#) to get you started. Both run on several .different platforms, including Windows, Mac OS X, and Linux

Keep software bug-free as an open source tester. You don't have to be a quality assurance .1 expert to get started with software testing. If you're using open source software already, you're already halfway there to being a tester. All it takes is sharing your experience if something doesn't seem quite right. Most projects have bug tracking applications that are easy enough for .even the novice user to report an issue or make a feature request

.Thrilled to see a new feature added to your favorite program? Don't forget to say [thank you](#)

Help others learn the ropes as an open source community leader. You can also help build .1 the open source movement by participating in a project's community. Many projects have their own dedicated forums or mailing lists, and there are plenty of spots on the net for participating more generally. Share you experience with others by helping answer questions from all sorts of users, from novice to expert. [Stack Exchange](#) hosts conversations around a variety of open .source tools

Write code as an open source developer. We say it a lot around here that the open source .1 way applies to much more than just software development. But code still matters! If you haven't tried your hand at programming yet, 2014 should be your year to get started. There are tons of great resources out there [for beginners to get started](#) and for fledgling programmers to further .[build their skillsets](#)

?Go here to learn to [program the open source way](#)

انتخابت را بردار

With so many different options for participating, anyone can join the open source community. How are you going to make 2014 your year of open source? Pick your role, and feel free to pick more than one! Need more guidance and inspiration? Opensource.com has these resources to .help you get started

[How non-programmers can contribute to open source projects](#) - [10 ways to contribute to an open source project without writing code](#) - [10 ways to start contributing to open source](#) - [10](#)

[ways to get started with open source](#) # کتاب های کلیدی برای خواندن
(<http://opensource.com/resources/ebooks>) منبع : [کتاب های متن باز]

ملزومات پروژه متن باز خیلی از نویسندگانی که تاریخچه وقایع متن باز را دارند و

شخصیت های مهم این جنبش از پرکارترین شخصیت ها هستند. اینها یک سری از مهمترین ها هستند. اگر ما یک کلاس درس داشتیم که اسمش "Open Source 101 بود پس این لیست درس ها بود

- [کلیسای اعظم و بازار](#) by Eric S. Raymond - [برنامه آزاد' جامعه آزاد] ()
- by Richard M. Stallman (/http://shop.fsf.org/product/free-software-free-society-2
تفریح [by Linus (/http://books.google.com/books/about/Just_for_Fun.html?id=6zSWd8Ou8BAC)
Torvalds and David Diamond - [کد باغی](#) by Glyn Moody - تولید با برنامه های متن باز ()
- by Karl Fogel (/http://producingoss.com آزادی برای همه ()
by Peter (/http://books.google.com/books/about/Free_for_All.html?id=gLxQAAAAMAAJ
Wayner - [آزادی به معنای آزادی](#) by Sam Williams - [متن باز](#) by Chris DiBona and Sam Ockman
- [بنایی برای شهر متن باز](#)

دنیای باز: داستان هایی از جامعه متن باز [منبع](#)] در opensource.com هدف ما تابش نور به جاهایی که از متن باز راهی هست که ایده ها را بزرگتر می کنه و تلاش ها را چند برابر. و علاقه ما همچنان در موضوعاتی مثل توسعه - تکنولوژی در حوزه هایی مثل کسب و کار' آموزش و پرورش' دولت' سلامت و قانون در زندگی روزمره باقی می مونه.
[این مجموعه]

(http://opensource.com/sites/default/files/open_source_eBook_AnOpenWorld_web.pdf) یک سری از بهترین داستان ها را به ارایه می ده. از کتابخانه! از خوابگاه' از پایتخت' از کلاس و اتاق جلسات رؤسا و دادگاه. اینها اعضای جامعه ما هستند. و شاید شما در نسخه بعدی باشید دریافت کتاب الکترونیکی بصورت مجانی یک دنیای باز: داستان هایی از جامعه متن باز [ePub] [PDF](#)] | ([http://opensource.com/sites/default/files/an_open_world.epub](#))
چگونه می توانید با متن باز شروع به کار کنید [لینک منبع](#) برداشتن قدم اول در یات مسیر می تونه ترسناک باشه. موانع جدید' مناظر نا آشنا برای عبور کردن و معمولا این مسیر بسیار رمزآلوده. اما این همان چیزیه که سرمایه گذاری جدید و هیجان انگیز را ارزشمند می کنه. مجموعه داستان هایی این کتاب الکترونیکی درباره ی کار کردن در مسیر متن باز (open source) است. چه به تنهایی به دنبال اطلاعاتی می گردی که از برنامه بسته فرار کنی' یا سازمان شما به دنبال یک برنامه یا یک راه حل می گردید برای ابزاری که بتونه به کسب و کار شما کمک کنه' اینها داستان هایی برای ورود به دنیای متن باز. چندان از این داستان ها مربوط به متخصصانی می شه که متن باز (open source) را کشف کردند' نشون می ده که چگونه شکوفا شدند از آن زمان. باقی آنها به کمک کردن برای پیدا کردن نرم افزار درست' استفاده از یک ابزار خاص یا معرفی اون به همسایه هاتون یا محلتون به قدرتی که در زبان باز (open source) هست تمرکز کردند.

این را به یاد داشته باش که کار کردن با یک چیز جدید اصلا آسون نیست' ولی' یک بودایی قدیمی می گه که بهتر خوب سفر کنی تا بعد بررسی. ما امیدواریم این داستان ها به شما کمک کنه تا به سلامت سفر کنید در مسیر متن باز و بتونید در سال های پیش رو سفر کنید. دریافت کتاب الکترونیکی بصورت مجانی چگونه می توانید با متن باز شروع به کار کنید [ePub] | [ODT](#)

باز همیشه برنده می شه : مجموعه Michael Tiemann [لینک منبع](#) Michael Tiemann نایب رئیس 'امور Open Source در Red hat و مدیر سابق Open Source در ابتکارات هست. اون که مقاله های زیادی برای Open Source نوشته' در مورد قدرت Open Source صحبت کرده و در بخشی از این پروسه ایده های خوب دیگران را برجسته یا هایلایت کرده.
از مقدمه خودش :

Linus Pauling معروف گفته "بهترین راه برای داشتن یک ایده خوب' داشتن ایده های متفاوته ."
به دلایل متفاوت این ساده ترین و بهترین گفته است. برای من' دلیل اصلی این هست که خیلی

ها نمی خواهند اونها را بخاطر ایده های خنده بشناسند' پس ما خودمون تغییرش می دیم. اگر ما خودمون شروع به تغییرش کنیم این باعث میشه که ما ایده های زیادی نداشته باشیم! و داشتن یک ایده خوب تقریباً غیر ممکن می شه!

چالش دوم بوجود آوردن یک فضا است ایده ها می تونند یکی 'ترکیب' تسريع بشند و تکامل پیدا کنند' اگر ایده های خوبی هستند شکل بگیرند. آیا این بهترین فضایی جدا شده است' جایی که ذهن های خلاق می تونن سنجیده بشن و بهترین ایده ها انتخاب بشن ؟ یا شاید این بهتره که پروسه ای را بوجود بیاریم که دیدگاه های متفاوت مطرح بشه' و بهترین نتیجه ممکن ازش بیرون بیاد ؟

خلاصه نگرش را از یک متن باز صحیح بدست بیار و یاد بگیر چطور این اصول را در کارت' شغلت' زندگیت' سرگرمی هات و چیزهای دیگه با این کتاب های دیجیتال اضافه کنی.

دریافت کتاب الکترونیکی بصورت مجانی باز همیشه برنده میشه [ePub] | [ODT]

اعمال اصول متن باز به دولت [لینک منبع] در opensource.com ما به تلافی متن باز (Open Source) و دولت ها نگاه میکنم' با توجه خاص به راه ها یی که دولتها تکنولوژی را در اختیار می گیرند و تکنولوژی های جدیدی در آزاد می کنند و ترویج جامعه open source ما در مسیری اصول مشارکت در open source مکتوب می کنیم که 'شفافیت' همکاری' به اشتراک گذاری' شایسته سالاری' جامعه و نمونه سازی سریع ارزشش فراتر از بخش تکنولوژی باشه. آزادی حرکت های مردمی در هر جا به رای دادن' پیگیری درخواست قانون آزادی اطلاعات (FOIA)' و حتی وضع قوانین (جمع سپاری) crowdsourced' نشان دهنده روند رشد اصول open source با دولت هاست.

این کتاب الکترونیک را دانلود کنید تا داستان های درباره امکانات دولت ها در سرار دنیا' از دو سال اول opensource.com بدست بیارید. آنها تاثیر متن باز را به دولت ها و برعکس نشان می دهند. هنوز راه طولانی باقی مونده و ما امیدواریم شما هم تجربه خودتان را با ما به اشتراک بگذارید.

دریافت کتاب الکترونیکی بصورت مجانی اعمال اصول متن باز به دولت [ePub] | [PDF]

خواندنی های مورد علاقه ی Opensource.com هر ساله تابستان 'جامعه OpenSource.com با هم همکاری می کنند برای لیست خواندنی ها. هر کدام از اینها پر از کتاب هایی است که توسط خوانندگان ما' حامیان و کارشناسان پیشنهاد شده تا منبع باز را بهتر و ساده تر احساس کنید. لیست ها را چک کنید تا خواندنی مورد علاقه تان را پیدا کنید.

- [لیست خواندنی ها سال 2010] (<http://opensource.com/life/10/8/open-books->) - [لیست خواندنی های سال 2011](#) - [لیست خواندنی های سال 2012](#) - [لیست خواندنی های سال 2013](#) - [لیست خواندنی های سال 2014](#) # مبانی متن باز

درباره این بخش این مجموعه از مقاله های برای معرفی مفاهیم کلی با متن باز و هک کردن طراحی شده.

مقاله ها - متن باز چیست - راهنمایی مقدماتی برای متن باز - شروع کار با Civic Hacking - چطور کار با پروژه متن باز را شروع کنیم - سه المان اصلی که همه پروژ های متن باز را تشریح می کنه - شروع کردن استفاده از گیت هاب - شروع کار با Discourse بسمت کد منبع باز در اندروید بوسیله اف دروید (F-Droid) چیزهایی که تازه وارد ها معمولاً درباره متن باز می پرسند کتاب های اصلی برای خواندن و سازمان های متن باز # Things newcomers to open source rarely ask but often wonder Source: [OpenSource.com](#)

[Open Source Comes to Campus](#) is an event series run by [OpenHatch](#) that introduces college students to open source tools, projects, and culture. Whenever we get a new-to-us question at an event, we write it down and answer it more fully on our blog. Here's a collection of "Infrequently Asked Questions" that are especially relevant for newcomers to open source

.projects

Q: I'm worried that I'll be a burden on a project because I'm so new. What kind of effort does a project have to make to build an open source community

A: First of all, you should know that expert open source community builders like Linus Torvalds understand this tension between training new contributors and the time it takes from existing maintainers to mentor them. I'll quote two paragraphs from [a 2004 post Linus made](#) on the Linux kernel list

:On Tue, 21 Dec 2004, Jesper Juhl wrote <
Should I just stop attempting to make these trivial cleanups/fixes/whatever patches? are they <
more noise than gain? am I being a pain to more skilled people on lkml or can you all live with
my, sometimes quite ignorant, patches? I do try to learn from the feedback I get, and I like to
think that my patches are gradually getting a bit better, but if I'm more of a bother than a help I
.might as well stop
:Linus replied <

To me, the biggest thing with small patches is not necessarily the patch itself. I think that <
much more important than the patch is the fact that people get used to the notion that they can
change the kernel—not just on an intellectual level (“I understand that the GPL means that I
have the right to change my kernel”), but on a more practical level (“Hey, I did that small
.change”)

Another thing to keep in mind that is, for most projects, the project leader is the only contributor
([source](#)). Consider this: The act of bugging them with questions as a newcomer can make them
!more enthusiastic about the project as a whole

If it were trivial to build an easily accessible open source community, virtually all projects
would do it. There is some effort involved. Of course, a lot of the work that makes a project
welcoming to newcomers—good documentation and setup guides, a well-maintained issue
tracker, active development, standards of conduct in the community—makes the project better
for everybody. But it does take time and energy, which many communities of projects aren't
willing to expend. That's why OpenHatch exists! We know that some projects put more effort
.into welcoming contributors, and we want to help you find those projects

The kinds of projects that welcome newcomers don't see you as a burden. They see you as a
huge help—even when you're struggling to understand bugs in the issue tracker or get the
development environment set up. When you ask a member of an open source project for help,
that gives them important information about what part of their project is confusing or incorrect.

The questions you ask and guidance you need lets them know how help others later on. And of
course, once you've gotten familiar with the project, you'll be able to help even more. The right
.kinds of project see that potential in you and want to work with you to get there

Two rules of thumb for people who worry about being too much of a bother (which includes us,
sometimes!): First, if people on the mailing list or the bug tracker haven't told you to stop
talking, then you're probably okay, even if they haven't replied to you yet. Second, if you want
.another opinion, just join the #openhatch IRC channel and ask us

Q: How do open source projects review changes, and how do those changes differ from a process like that of Wikipedia

A: One key cultural element of Wikipedia, at least so far in its history, is that anyone's edits to
English Wikipedia immediately become part of the live version of the encyclopedia. By

contrast, to maintain quality, open source projects typically permit only a handful of people to *.directly merge* changes into the main project. Therefore, submitted changes go through review. Different communities have different standards and processes. Sometimes, there is no review at all, just automatic merging by the maintainer. In other cases, like Linux, submissions go to a mailing list [for review](#). Others, like the OpenHatch web app, use web-based tools like GitHub pull requests. A great reference for a more complicated process is the [OpenStack Gerrit Workflow documentation page](#).

?Q: How do you make time to contribute to open source when you're a busy student

A: By simply using open source programs, you are uniquely positioned to help other users. Don't forget that helping people use the software is a substantial contribution to the community!

By becoming an expert in whichever programs you use, you will also gain the knowledge to help convert bugs filed by others into actionable reports for the main developers. And once you reach that point, you might find it easier to just fix the issue

That said, it can be daunting to add another activity to schedules already stuffed full of classes, paid work, and student life. One way to approach open source is to see it not as an alternative to doing schoolwork but as part of your education. For computer science majors, open source projects are a great way to practice the concepts you're being taught in class. In the other sciences, learning how to use and contribute to open source tools such as R or Octave (or ImageJ or PsychoPy or JMARS) can help you excel when doing laboratory work—and it looks great on your resume. Even in the arts and humanities, learning about open source tools such as

.Processing can help you succeed. For students who work while they're at school, open source projects can be potential employers.

You can apply for [Google Summer of Code](#) or the [GNOME Outreach Program](#) and do a paid internship at an open source project. Individual companies working with open source may hire students that show enough interest and ability. You can even employ yourself using open source—for instance, you could do freelance work making websites and applications by using open source tools like WordPress and Drupal

You can make open source a part of your social life as well. Join (or start) a Students For Free Software chapter on your campus or invite friends over for a 'bug-fixing party'

Finally, remember that your contributions to open source can be as big or as small as you want them to be. If you only have time to spend a few hours a month writing documentation or fixing small bugs in a project, that's great—the skills you're learning and connections you're making will serve you well if you ever decide to get more involved

Q: What is the difference between source downloads for developers and downloads for users? What's a stable release

A: Open source projects often release the program in a few different ways. Typically, if you are trying to contribute, you will want the latest source code cloned or 'checked out' from the project's version control system. If you are just trying to use the program, you typically want a user-oriented download

Many programs' source code can't be executed directly on a computer—it must be 'compiled', which turns it from human-readable text into machine-executable binary. The most common languages where this is true are Java, C, and C++. Therefore, the most useful download for a person who wants to run a project like OpenOffice, which is mostly written in C++, would be a compiled version specific to their operating system (for example, a Windows-specific build). By

contrast, if you want to contribute to the program, you should act similarly to how the main developer acts—that is, by using a version control system on the editable program text, the .source code

Another important concept is the stable release. During development, people make contributions that break the program, conflict with other changes, or are simply unfinished. At any given point the most recent version might be completely unusable. So, maintainers periodically work towards releases (noun), which are tested to make sure they're usable. If they are, maintainers .release (verb) it

So if you're a user, you'll want the stable release most appropriate to your operating system and .computer. If you want to contribute, you'll want the latest, non-compiled version

Q: Are all open source projects welcoming to newcomers? How can you tell when the ?communities around a project are filled with jerks

A: Not all open source projects are welcoming to newcomers. Not all open source projects need to be. Some projects benefit from keeping their communities small and experienced. And, some .maintainers simply don't have the energy or the inclination to mentor new people

There's nothing wrong with that, but it's a shame when newcomers try to get involved with a project that isn't interested in involving them. It's a frustrating experience for everyone and .unnecessary when there are plenty of people who'd be excited to work with them

So how can you find a good project for you, a newcomer, to contribute to? Here are some good :signs to look for

A large, active community is more likely to have members who can take the time to mentor. It - also means that contributions will be acted on more quickly. - Good documentation and demos mean that developers have thought about how to introduce people to their project. - Some projects have [Codes of Conduct/Diversity Statements](#), which demonstrate that the community is trying to be a safe and welcoming space. - Projects that are part of [Google Summer of Code](#) or the [GNOME Outreach Program for Women](#) have made a commitment to being good .environments for newcomers

At OpenHatch, we try to identify projects that are especially good for newcomers. We're also developing a list of [OpenHatch-affiliated projects](#) that are working with us to make contributing to their projects as easy as possible. (If you'd like to be on that list, let us know!) Feel free to contact us and ask us for recommendations or if a particular project is known to be good for .newcomers

Of course, there's a difference between not having the time and energy to help newcomers and being actively mean or even abusive about it. If you experience the latter and our comfortable telling us about it, we'll do our best to support you. OpenHatch people are always willing to provide advice on projects to join, either to help you find a great experience or to make the most of a bad experience. Find us on IRC (#openhatch on irc.freenode.net) or email us at hello@openhatch.org. You can also join groups such as [Syssters](#) or the [Empowermentors](#) .[Collective](#), which may help you navigate the free software community

More resources ##

This article has been a collection of information gathered during Open Source Comes to :Campus events

[Infrequently Asked Questions: Wellesley College](#) - [Infrequently Asked Questions: UMass - Amherst](#) - [Infrequently Asked Questions: Bloomington](#)

.Read more posts about this [event series](#) on our blog
For blog posts about getting involved with OpenHatch, read about the experiences of two of our contributors: [Britta](#) and [Mandar](#). # Three key elements that define every open source project

Source: [OpenSource.com](#)

Open source has come a long way in the past 30 years and is entering the consciousness of most modern cultures. When thinking of open source projects, people categorize them several ways: governance structure, type of product platform, programming language, utility, technical details (language written in), industry sponsored or fully independent, and more

But what truly defines any open source project, making it a unique entity different from all other open source projects? I would propose that there are three key elements of any open source project that frame, define, and differentiate that project from all others: the code, the community, and the brand

The code ##

Code is king. Code is what makes a product do something, and that's why open source projects exist in the first place: to build something useful. Technologists get excited about what the code does, and how it does what it does. Marketers get excited about how the product will solve their customers' problems. Code is what most people are looking for when they're searching for an open source project to use

Sounds simple enough—so why don't we define an open source project purely based on its code? As anyone who has worked in software development already knows, code is ever-changing and ephemeral. In the open source frontier, free from the traditional controls of corporate-led projects, "the code" can get very hard to follow: open source code is infinitely forkable. Once your code is checked in under an [Open Source Initiative \(OSI\) license](#) to a public repository, it is fully accessible to anyone and everyone to take—and modify—for their own purposes. Once another user forks the code from your project and makes a slight modification, it is no longer officially part of your original project

The community ##

If code is the "what" of a project, then the community is the "who" of the project—the people who make it all happen. The core community of a project includes anyone actively involved in moving the project forward, such as the engineers writing the code and the end users who provide feedback or request specific modifications. The overall community also includes people who don't check code, but provide support such as governance/process oversight, public relations/marketing, training, or financial or employment support. The social norms, the etiquette, and the ethos of the community help to differentiate a project from all others

While participation in an open source project may be part of some individuals' paid employment (e.g. a corporate-employed software engineer assigned to work on an open source project for a certain percentage of her or his time), most open source community members participate voluntarily with no direct connection to their paycheck. So members tend to come and go as their interest or their other commitments wax and wane, or as their employer changes strategy.

.Like the code, the community is ever-changing

Unlike a corporate software development project that can plan on having employees with certain skills available to assign to do specific work, the participation in an open source community is unpredictable and often beyond the control of the project. Personality conflicts arise and may result in highly skilled contributors leaving the community more readily than they

would leave a paid employment. But the benefits of an open community can be seen in the enthusiasm and drive of many community members, and in the longevity of successful project communities, and in synchronicity and great work forward on the code

The brand ##

Brand is how the world outside of an open source project learns about that project. When individuals or companies are deciding which project to use or to invest in, branding helps them differentiate projects that offer similar functionality. Of course they will consider other details, but it is much easier to think, "Do I want to support Hadoop, with the yellow elephant?" rather than "Do I want to support Cloudera's CDH or the Hortonworks Data Platform, or the newly announced ODP?"

The brand" includes many things: the official name of the project, a logo for the project or product, and even the appearance of the project's website and your product's UI. Some branding components in particular are legal trademarks: these typically include the official software product name and logo, although trademarks are strongest when used consistently

Unlike code and community, a project's brand is not ever-changing or ephemeral. A trademark cannot be forked without legal permission, and a project brand can stay consistent even when community membership fluctuates. In many ways, the brand and trademarks are the element of a project that can be most easily controlled and maintained. However, proper trademark usage can be overlooked—or under appreciated by the community inside of the project—as an important tool for defining a project's unique character. Given that anyone can fork the code,

and that community members come and go, a project's brand and trademarks are crucial elements for maintaining a project's longevity and independence, and to continue to draw in new project contributors (<http://opensource.com/resources/what-open-source>):

اصطلاح "منبع باز" به چیزی گفته میشود که بنا به طبیعت ساختش قابل ویرایش است. در حالی که اصطلاح منبع باز در مورد ساخت و طراحی نرم افزارهای رایانه ای کاربرد دارد، امروزه اصطلاح "منبع باز" اشاره به مجموعه ارزشهایی دارد که ما به آن *طریقه منبع باز* میگوییم (<https://opensource.com/open-source-way>). پروژه ها، محصولات، و یا طرح هایی منبع باز هستند که مبادله باز، همکاری مشترک، تعدد مدلهای آزمایشی، شفافیت، شایسته سالاری، و ساخت انجمن را ارزش می نهد. برنامه با کد منبع باز چیست؟

برنامه با کد منبع باز به برنامه ای گفته میشود که کد منبع آن برای ویرایش و بهبود سازی در اختیار عموم باشد.

"کد منبع" در حقیقت قسمتی از برنامه است که کاربران آن را نمی بینند؛ قسمتی که برنامه نویسان میتوانند دستکاری کنند تا کارکرد قسمتی از برنامه عوض شود. برنامه نویسانی که به کد منبع برنامه ها یا نرم افزارها دسترسی دارند می توانند با اضافه کردن خصوصیت های مختلف برنامه را بهتر کنند و یا قسمتهایی از برنامه که همیشه بدرستی کار نمیکند را تصحیح کنند. فرق برنامه های با کد منبع باز با بقیه انواع نرم افزارها چیست؟

کد منبع بعضی از نرم افزارها فقط بوسیله شخص، گروه و یا شرکتی که آنها را ساخته و پشتیبانی میکند قابل تغییر است. بخاطر داشتن ملکیت این نرم افزارها به طراح و سازنده و اینکه تنها آنها هستند که قانوناً میتوانند نرم افزار را تغییر یا کپی کنند، این نوع از نرم افزارها را "نرم افزارهای اختصاصی" یا "نرم افزارهای با کد بسته" می نامند. نرم افزارهایی مانند آفیس ورد یا ادوبی فوتوشاپ از این دسته هستند. برای استفاده از نرم افزارهای اختصاصی معمولاً کاربر باید بپذیرد که بدون اجازه مالک، نرم افزار را تغییر ندهد (معمولاً با قبول عهد نامه ای که در ابتدای نصب نرم افزار نمایش داده میشود).

نرم افزارهای با خصوصیت کد منبع باز متفاوت هستند. سازندگان آنها کد منبع را در دسترس عموم قرار میدهند (<https://opensource.com/business/13/5/open-source-your-code>) تا بقیه بتوانند آن را مشاهده، کپی، تغییر، به اشتراک و یا مطالعه کنند. نرم افزارهایی مانند لیبرآفیس (<https://www.libreoffice.org>) یا گیمپ (<http://www.gimp.org>) از این نوع برنامه ها هستند. همانند نرم افزارهای اختصاصی، کاربران باید عهد نامه ای را در ابتدای نصب نرم افزار بپذیرند، اما بند های تعهدنامه نرم افزارهای با کد منبع باز تفاوت فاحشی نسبت به نرم افزارهای اختصاصی دارند. تعهد نامه نرم افزارهای با خصوصیت کد منبع باز همکاری و با اشتراک گذاشتن را ترغیب میکنند و به افراد اجازه میدهند که کد منبع آنها را تغییر داده و این تغییرات را در پروژه های خود بکار گیرند(مثال - <https://opensource.com/law/13/5/does-your-code-need-license>). بطوری که اگر کسی کد منبع نرم افزار کد منبع بازی را تغییر داد و نرم افزار را با بقیه با اشتراک گذاشت باید کد منبع آن نرم افزار را هم بصورت رایگان در اختیار همگان قرار دهد. به تعبیری دیگر برنامه نویسان کامپیوتری میتوانند نرم افزارهای با کد منبع باز را تغییر و یا مرور کنند تا وقتی که به بقیه اجازه چنین کاری برای کد تغییر داده خود بدهند. در واقع اگر این اجازه را ندهند ناقض تعهد نامه نرم افزار کد منبع باز خواهند شد.

So as the [Open Source Initiative](#) explains, "open source doesn't just mean access to the source code." It means that anyone should be able to modify the source code to suit his or her needs, and that no one should prevent others from doing the same. The Initiative's definition of "open source" contains several other [important provisions](#).

?Is open source software only important to computer programmers ##

Open source software benefits programmers and non-programmers alike. In fact, because much of the Internet itself is built on many open source technologies — like [the Linux operating system](#) and the [Apache Web server application](#) — anyone using the Internet benefits from open source software. Every time computer users view webpages, check email, chat with friends, stream music online, or play multiplayer video games, their computers, mobile phones, or gaming consoles connect to a global network of computers that routes and transmits their data to the "local" devices they have in front of them.

The computers that do all this important work are typically located in faraway places that users don't see or can't physically access—which is why some people call these computers "remote computers." More and more, people rely on remote computers when doing things they might otherwise do on their local devices. For example, they use online word processing, email management, and image editing software that they don't install and run on their personal computers. Instead, they simply access these programs on remote computers by using a Web browser or mobile phone application.

Some people call remote computing "cloud computing," because it involves activities (like storing files, sharing photos, or watching videos) that incorporate not only local devices, but also the global network of remote computers that form an "atmosphere" around them. Cloud computing is an increasingly important aspect of everyday life with Internet-connected devices. Some cloud computing applications, like Google Docs, are closed source programs. Others, like [Etherpad](#), are open source programs.

Cloud computing applications run "on top" of additional software that helps them operate smoothly and effectively. The software that runs "underneath" cloud computing applications acts as a [platform](#) for those applications. Cloud computing platforms can be open source or closed source. [OpenStack](#) is an example of an open source cloud computing platform.

?Why do people prefer using open source software ##

Many people prefer open source software because they [have more control](#) over that kind of software. They can examine the code to make sure it's not doing anything they don't want it to do, and they can change parts of it they don't like. Users who aren't programmers also benefit from open source software, because they can use this software for any purpose they wish—not .merely the way someone else thinks they should

Others like open source software because it helps them [become better programmers](#). Because open source code is publicly accessible, students can learn to make better software by studying what others have written. They can also share their work with others, inviting comment and .critique

Some people prefer open source software because they consider it more [secure](#) and stable than proprietary software. Because anyone can view and modify open source software, someone might spot and correct errors or omissions that a program's original authors might have missed. And because so many programmers can work on a piece of open source software without asking for permission from original authors, open source software is generally fixed, updated, and .upgraded [quickly](#)

Many users prefer open source software to proprietary software for important, long-term projects. Because the source code for open source software [is distributed publicly](#), users that rely on software for critical tasks can be sure their tools won't disappear or fall into disrepair if .their original creators stop working on them

?Doesn't "open source" just mean something is free of charge ##

No. This is a [common misconception](#) about what "open source" implies. Programmers can charge money for the open source software they create or to which they contribute. But because most open source licenses require them to release their source code when they sell software to others, many open source software programmers find that charging users money for software services and support (rather than for the software itself) is more lucrative. This way, their software remains free of charge and they make money helping others install, use, and .troubleshoot it

? "What is open source "beyond software ##

At opensource.com, we like to say that we're interested in the ways open source can be applied to the world beyond software. We like to think of open source as not only a way to develop and license computer software, but also an attitude. Approaching all aspects of life "[the open source way](#)" means expressing a willingness to share, collaborating with others in ways that are transparent (so that others can watch and join too), embracing failure as a means of improving, .and expecting—even encouraging—everyone else to do the same

It means committing to playing an active role in improving the world, which is possible only when everyone has access to the way that world is designed. The world is full of "source code"—[blueprints](#), [recipes](#), [rules](#)—that guide and shape the way we think and act in it. We believe this underlying code (whatever its form) should be open, accessible, and shared—so .many people can have a hand in altering it for the better

Here, we tell stories about what happens when open source values are applied to business, [education](#), [government](#), health, law, and any other area of life. We're a community committed to telling others how the open source way is the best way—because a love of open source is just .like anything else: it's better when it's shared

?Where can I learn more about open source ##

We recommend visiting the Opensource.com [resources page](http://opensource.com/resources/what-open-source)
اصول طراحی آندروید منبع : <http://opensource.com/resources/what-open-source>
Android.com

این اصول طراحی توسط تیم کاربران تجربه ای اندروید برای بهترین نحوه استفاده کاربران اندروید تهیه شده. یکی از دلایل طراحی آن این است که دستورالعمل های مربوط به طراحی جزئیات برای دستگاه های مختلف اندروید را برای توسعه دهندگان و طراحان مشخص کند. این اصول را همزمان با خلاقیت و نگاهت برای طراحی در نظر بگیر. کارت را انجام بده ولی برای انجام دادنش دلیل داشته باش.

منو فریب بده / منو جادو کن ### منو از طریق راه های تعجب آور شوق زده کن یک دستگاه زیبا که انیمیشن خوبی به همراه افکت های صوتی که خیلی خوب زمانبندی شدند می تونه چیزه خوبی برای امتحان کردن باشه. این اثرات کوچک هستند که می تونه تاثیرات بسزایی برای امتحان کردن داشته باشه و به بیانی حس اینکه نیروی قدرتمندی در دستاته. ### اجسام واقعی می تونن از دکمه ها و منو ها باحال تر باشند به همه اجازه بده بطور مستقیم همه چیز را در برنامه ات دستکاری کنند. این زحمت را برای شناخت کمتر می کنه در حالی که از ساختنش احساس راضی بودن خواهی داشت.

بزار اون را مال خودم کنم همه دوست دارن بتون دستکاری های خودشون را انجام بدن ' چرا که این کمک می کنه احساس کنن همه چیز تحت کنترل خودشونه و به بیانی حس کنن تو خونه خودشون هستند. پیش فرض خوب و معقول را حتما قرار بده ولی به این فکر کن که باحال باشن ' همچنین تغییرات سفارشی که جلوی انجام کارهای اولیه را نگیره.

منو بهتر بشناس اون چیزی که مردم ترجیح می دهند را در طول زمان بفهم . به جای اینکه مکررا از اونها بخواهیم همون انتخاب قدیمی را دوباره و دوباره انجام بدن انتخاب های قدیمی را جایی بزار که بتونن به راحتی بهش دسترسی داشته باشند.

زندگی من را آسانتر کن. ### کوتاه نگهش دار از عبارات کوتاه به همراه کلمات آسون استفاده کن. مردم جملات دوست ندارن مخصوصا اگر بلند باشند.

عکس ها از کلمه ها سریعتر هستند سعی کن از عکس ها استفاده کنی برای توضیح دادن ایده هات. عکس ها توجه همه را بخودشون جذب می کنند و می تونند تاثیرگذارتر از کلمات باشند.

برای من تصمیم گیری کن ولی بزار حرف آخر را من بزنم بهترین حدس را بزن و عمل کن بجای اینکه بخوای اول سوال کنی . انتخاب های زیاد م تصمیم گیری ها همه را ناراحت می کنه. فقط برای اینکه اگر اشتباه کردی ' امکان بازگشت را باز بزار.

فقط چیزی را بهم نشون بده وقتی که بهش نیاز دارم مردم وقتی همه چیز در یک لحظه را با هم می بینند درونش غرق می شوند (دچار اشتباه می شوند) . اطلاعات و داده ها را به بخش های کوچک و قابل هضم تبدیل کن. قسمت های غیر ضروری را در حال حاضر پنهان کن و به مرور زمان به کاربرات یاد بده.

من باید همیشه بدونم کجا هستم به همه این امکان و اراده را بده که بدونن کجا هستن. مکان هایی در برنامه ات بوجود بیار که ظاهری متفاوت داشته باشه و از تغییراتی استفاده کن برای نشان دادن رابطه میان صفحه نمایش. همچنین داشتن امکان پیشنهاد و انتقاد باعث پیش رفت می شه.

هیچوقت چیزهایم را گم نکن اون چیزی را که کاربرات وقت گذاشتن و درست کردن ذخیره کن و بهشون اجازه بده از بهش دسترسی داشته باشند از هر جایی. تنظیمات و دستکاری های شخصی را پیاد داشته باش و نوآوری ها جدید را در موبایل ' تبلت ' کامپیوترها . این بروزرسانی را یکی از آسانترین کارهای دنیا می کنه.

اگر همون شکلی باشه ' همنجوری هم کار می کنه به مردم کمک کن برای تشخیص دادن تفاوت عملکردها با ساختن اونها با شکلی متفاوت بجای اینکه ظریف باشند. اصول و سبک را

کنار هزار چرا که می تون بصورت ظاهری یکی باشند ولی عملکردی متفاوت داشته باشد
 ## فقط وقتی مزاحم بشو که خیلی مهمه. مثل یک دستیار شخصی خوب باید سپری باشی
 در مقابل جزییات غیر ضروری. مردم می خواهند متمرکز بمونن ' قبل از اینکه خیلی جدی باشه
 و یا زمان خوبی نباشه مزاحم شدن می تونه واقعا سنگین و خسته کننده باشه.
 ## منو شگفت زده کن ### به من راه کاری را بده که همه جا کار می کنه مردم احساس
 خوبی می کنند وقتی خودشون می تونن مشکلشون را حل بکنند. برنامه ات را آسان کن با
 اعمال نفوذ از الگوهای تصویری و خاطره کار کردن با دیگر برنامه های اندروید. به عنوان نمونه
 حرکت جابجایی مثل جاروب کردن می تونه یک میانبر خوب باشه برای جهتنامی.
 ## این تقصیر من نیست مراقب باش که چگونه به کاربرانت بگی که تغییرات را انجام بدن.
 اونها دوست دارند احساس هوشمندی کنند وقتی دارند از برنامه ات استفاده می کنند. اگر
 مشکلی بوجود آمد ' بهشون راه حل را نشون بده ولی دستور العمل بازیابی فنی را براشون
 بزار. اگر خودت می تونی مشکل را حل بکنی بدون حضورت (پشت پرده) می تونه خیلی بهتر
 باشه.

کم کم تشویقم کن مراحل پیچیده را به بخش های کوچکتر تبدیل کن که به راحتی بتونن
 انجام بشن. در هر مرحله پیشنهاد بده حتی اگر خیلی حساس و ظریف باشند.
 ## بلند کردن سنگین را انجام بده / کار سخت را برای من انجام بده به تازه کارها با در
 اختیار گذاشتن امکان انجام کارهایی که قبلا فکر می کردند نمی توانند انجام بدن احساس
 متخصص بودن بده. به عنوان مثال با میانبری از چند افکت تصویری می شه یک عکس
 معمولی را به یک عکس حرفه ای تبدیل کرد اون هم فقط با انجام چند حرکت.
 ## کارهای اصلی را سریعتر کن همه حرکات با هم یکسان نیستند. تصمیم بگیر که چه چیزی
 از همه مهمتر می تونه باشه در برنامه ات برای پیدا کردنش و استفاده سریع! مثل دکمه شاتر
 دوربین برای عکس گرفتن ' و یا دکمه نگه داشتن موزیک در یک موزیک پلیر. - آیا واقعا
 طراحی بطور متن باز (open source) امکان پذیره ؟

منبع: [OpenSource.com](https://opensource.com)

A few weeks ago, I wrote an article [about Apple and open innovation](#). The discussion in the
 comments about Apple's success, despite their non-openness, was pretty interesting. [Greg](#)
[DeKoenigsberg](#) started things off with this salvo

No community could build something as gorgeous as the iPhone; it requires the singular " <
 vision of a beautiful fascist, and the resources of a gigantic company, and a world full of users
 ".who would happily trade simplicity and certainty for the ability to tinker

I think few people would argue that one of Apple's greatest strengths is their amazingly
 consistent, and consistently beautiful, design work. And when I say design, I mean both "little d
 design" (their stuff looks awesome) and "big D Design" (their systems, processes, and
 experiences are expertly rendered)

From a design perspective, Apple has figured out how to make lightning strike in the same place
 over and over again

امروز می خواهم سوالی بپرسم که مدتی بود داشتم بهش فکر می کردم:
 - آیا واقعا طراحی بطور متن باز (open source) امکان پذیره ؟

Meaning, can a group of people designing collaboratively, out in the open, ever do the kind of
 consistently beautiful design work that Apple does? Or is Greg right, that "no community could
 ?"build something as gorgeous as the iPhone

Both of my partners at [New Kind](#), [David Burney](#) and [Matt Muñoz](#), are designers by
 background. Both of them have significant open source experience (David spent almost 5 years
 as the VP of Communications at Red Hat, Matt worked on many Red Hat projects, including

.designing the Fedora logo), so the three of us have talked about this subject many times before. Even on the Fedora logo project, which was one of our first experiments in open design (learn more about the project and process [here](#)), only the input and feedback part of the process was open. Ultimately the logo itself was designed by one person (Matt), with a lot of feedback along the way from a very engaged community.

I've spent time looking at "crowdsourced design" companies like [99designs](#) and [Crowdspring](#). Many folks have written about companies like this as the future of design ([great post](#) last week by Hutch Carpenter on crowdsourcing and the disruption of the design industry). And old school designers are scared to death that these companies will commoditize the design industry and put them out of work. Maybe. But I'm not so sure.

Spend a few minutes on one of these sites and you'll see almost all of the projects are simple "little d design" projects: websites, t-shirts, icons, logos, that sort of thing. I didn't see a big market for experience design or other more complex "big D Design" activities. These sites don't seem very collaborative to me either. Sure, you can see the designs that other users have submitted. It is transparent. But you are not incented to collaborate with other designers, almost the opposite, since only one design wins the money.

So tell me. Will great design ever be a bazaar, or will it always be a cathedral? Do you know of examples where amazing design work—big D or little d—is being done the open source way? Do you think open, collaborative design will ever be as good as (or better than) design done the traditional way?

مطمئن نیستم.
دوست دارم افکار تو رو بدونم و بشنوم. # Five steps to using design in your open source project

Source: [OpenSource.com](#)

At the [Open Technology Institute](#) (OTI), we've been working on opening our user feedback process as a way to improve our internal processes and collaboration, engage our user community more, promote non-developer contributions, and think more broadly about how open source process plays a role in the [Commotion Wireless project](#), a free and open-source communication tool that uses mobile phones, computers, and other wireless devices to create decentralized mesh networks.

.More on that in [my previous article](#)

In this article, I've expanded on the five tips I've identified for leveraging user-centered design in your open source project.

.1 Track usability and design issues

Gather context as well as feedback - Don't respond (with software/design changes) to each piece of feedback - Synthesize issues for themes

As a result of our field activities, we had a pretty large queue of feedback from trainers, trainees, and users on documentation, usability and design, feedback from hackathons, the developer and discussion mailing lists, as well as a tracker for issues that people submit via the website. Given the amount of feedback we had gathered over the last few years, we decided to start by synthesizing and categorizing the themes in that dataset.

Understanding user feedback can be very challenging. Many developers and designers hear feedback or read notes from a user feedback session, and want to make changes to the user interface based on each comment a user makes. What a user says exactly is not always the

change that you need to make—it's important to understand the context in which the user provided the feedback. We've run into this a few times on the Commotion project, as much of our feedback comes from hands-on training sessions with novice users

The main problem here is that it becomes difficult to understand if the feedback is on the documentation that the user is following, something confusing in the interface design or a mismatch between the documentation and the interface. Understanding the context and the user motivations and goals, and documenting that with the actual feedback—observational or commentary—is crucial for assessing the feedback later. Not to mention that it makes it possible for other team members to understand the context without having been present

.1 Understand who you want to design for

When making design decisions, you often have to prioritize which type of user to design for—novice or expert users. Our goal with the Commotion project is that anyone should be able to set up a mesh network, which would push us in the direction of designing for novice users

The challenge is that in network administration and management, users will frequently and sometimes very quickly need access to the more advanced settings and configuration tools.

Striking the balance between novice usability and expert access can be challenging, and it's important here to remember that each round of usability improvements is just that - the current round or iteration. You might not be able to fix everything this time, and you might not have even found all of the issues to fix

The important thing is that you focus on keeping it simple and learnable to allow novice users to find their way around and expert users the ability or doors to get to the features they know they need

.3 Make prototypes

Leverage the style of the prototype (low-fidelity vs high-fidelity) to get the type of feedback you need

Similar to the issue I described earlier about understanding the context surrounding the user feedback, it's important to understand the context you are creating with your prototypes. High-fidelity prototypes or mockups are a great way to get detailed design/visual feedback from users—e.g. about colors, language, placement of buttons, etc. High-fidelity prototypes are not a good way to get feedback about the logical process flow—e.g. does your software match users' mental models of how the software should work

The basic rule of thumb here is that the sketchier (literally more sketch-like) your prototypes are—the more likely the feedback will be about information structure and process flow—i.e.

challenging your basic design assumptions, whereas the more detailed the prototypes are the less likely the user is to challenge it because they have a sense that you've already finished and so focus on the aesthetic issues. The type of prototype you make should be based on the type of feedback that you need

.4 Understand how the software works from the user's perspective

The next step of the process for us was to reach out to user groups—some whom we had trained and some who had adopted the software on their own to have them do a walkthrough of the prototypes. We looked for novice users as well as expert users to make sure that the changes we were making wouldn't break anyone's mental models of the software and to make sure that we were burying any tools that needed to be top level in the interface. We scheduled as many users as we had time to work with, and used screen sharing software to walk through some standard

.tasks using Commotion

To do this, we prepared a plan for our interactions with the user community, asking them to Think Aloud through some of the common tasks such as configuring a wireless router, checking on the network status, finding out how many users are currently connected, and things like that. We also had some standard "demographic" questions to get a sense of the user's context, such as their familiarity with the software and with managing networks, what type of network they were involved in, their role within the network, and what type of operating systems and technology they use most often. All of this information can help your team in building better user profiles or personas and can feed into more informed design processes for new features down the line

Use usability/design iterations to build your community: hackathons, mailing lists, and distributed tools

Similar to what the [Magnolia CMS team described in their post about designing for mobile versus desktop](#), it's important to be open and transparent about the usability review and design process as it can help you engage your broader community. This means cultivating conversations on mailing lists, encouraging constructive criticism of UX and design, or even asking current members to recruit others that work in UX to the project

For the Commotion team, this was just one iteration of what has been and will continue to be many improvements to the user interface, but it gave us a chance to practice our process, catch up on a backlog of user interface issues, and work on developing our community. It also gave us an opportunity to show the community the many ways that they contribute to the process, aside from just contributing code. We're making sure that whenever we go to a hackathon, we have our documentation ready to be hacked on along side our software, website, and everything else, so that we can keep making it better and easier to use

اعمال نفوذ طراحی کاربر محور در پروژه متن باز

منبع: OpenSource.com

وقتی که من در Open Technology Institute شروع به کار کردم (<http://oti.newamerica.net>)

(OTI)، همیشه این سوال را از خودم می پرسیدم: چرا طراحان UX علاقه دارن در یک مجموعه متن باز کار کنند؟ واقعیت در نظر من اینه که همه طراحی ها و قابلیت ها بطور طبیعی با متن باز کار می کنند.

طراحان' طرح ها و ایده ها را از الگوهای طراحی می گیرند که ما تجربه کردیم و یا در دنیای اطراف ما هستند. ما بطور مداوم ایده هامون را از مفاهیم متفاوت با هم به را آزمایش و یا به اشتراک می زاریم تا طراحی بهتر قابل درک بشه. منابع خیلی خوبی برای قابلیت های استفاده و بهترین روش های طراحی در اینترنت هستند (پایین را ببینید) و خیلی از کاربران متن باز از این چارچوب به عنوان اصولی در جعبه ابزار های خودشون اضافه می کنند' پس خط بین توسعه دهندگان' طراحان و غیره خیلی مبهمه.

اینجا یک سوال باقی می مونه: چرا ما به ندرت قابلیت های استفاده حرفه ای و طراحان را در جامعه متن باز می بینیم' و ما چطور می توانیم شروع به تغییر این کنیم؟

غالباً پروسه ها و پروژه های زبان باز در طراحی و مدارس UX و کنفرانس ها برجسته نیستند' محدود کردن آگاهی ها در جوامع طراحی از نیاز های این حوزه است. علاوه بر این' همون طور که در بالا اشاره کردم' سیر تکامل نرم افزارها' چارچوب ها' کتابخانه ها و جعبه های ابزار مثل Twitter Bootstrap و AngularJS اجازه می ده که وظیفه ها کمرنگتر بشند و این ساختن را آسان تر می کنه. همزمان با اینکه در یک دست دانه مشخص میشه که قابلیت های استفاده و طراحی نیاز های حیاتی هستند که هم برای کاربران و هم برای توسعه دهندگان پروژه ها (APIs) هم نیاز به استفاده است)' ظاهراً در استفاده از نرم افزار های زبان باز یک شکاف درصدی

وجود دارد. در گذشته، پروژه های کوچک تجربه کاربران را به عنوان یک چیز تجملی مشاهده می کردند (UX) و بنابراین از نیروی انسانی برای نیازهای UX سرمایه گذاری نمی کردند. همچنین من می توانم تصور کنم که تقاضای بازار از طراحان را برای کار کردن در کمپانی های های خوب پول می دن اونها را نگه داشته، در حالی که بسیاری از پروژه های زبان باز یا بصورت مجانی اداره می شن یا بودجه خیلی کمی دارند. به هر حال، من اعتقاد دارم که یکی از دلایلی که ما طراحان و قابلیت های استفاده حرفه ای را در جامعه متن باز نمی بینیم این هست که اونها مطمئن نیستن که چطور می توانند در پروژه ها همکاری کنند یا در گذشته در نتوسن خودشون را در پروژه های متن باز نشان بدهند. در Open Technology Institute (OTI)، ما داشتیم به این فکر می کردیم که چطور راهی را پیدا کنیم برای استفاده فیدبک کاربران در بهتر شدن در تغییرات داخلی و همکاری ها، کمک بیشتر کاربران در جامعه، ترویج همکاری کسانی که برنامه ساز نیستن و بزرگتر فکر کردن به اینکه جایگاه پروسه زبان باز در (پروژه چمدان باز) (<http://commotionwireless.net>)، یک ابزار متن باز برای ارتباط که از موبایل ها، کامپیوترها و دیگر دستگاه های سیار برای بوجود آوردن شبکه های مش (تور) استفاده کنه.

برای (نسخه رسمی 1 پخش شده) (<http://commotionwireless.net/blog/commotion-router-v1-release-notes>) نرم افزار ما تصمیم گرفتیم به یک تغییرات کلی روی رابط کاربری، اولین تمرکزمان به روی نسخه روتر سیار بود و همچنین هماهنگی سایر سیستم عامل ها (Mac, Linux, Android, Windows, iOS) با روتر. سفت افزار شبکه مش چیزی نیست که هر روز باهاش سروکار داشته باشید - تیم توسعه دهندگان ما غالبا تمرکزشون روی اضافه کردن قابلیت های بیشتر هست در حالی بطور همزمان تلاش می کنند که فایل ها را کوچک تر کنند تا بتونند در سخت افزار که جای خیلی کمی دارند جا بشند (فقط 5.2 مگابایت). همزمان تیم حوزه ما که منم بخشی از اون هستم داریم تحقیق می کنیم که چطور قدرت نفوذ شبکه مش را در روابط اجتماعی، آموزش توسعه و اتخاذ چارچوب و امتحان کردن این ایده ها را با جامعه ای که تجربه کردن را با جامعه های که زیرساخت هاش متعلق به خودشون بالا ببریم. در مقاله بعدیم (<http://opensource.com/life/14/2/Five-steps-using-design-open-source-project>)، من این 5 نکته ای که یاد گرفتیم را از بررسی های اعمال نفوذ کاربرا در اینترفیس بهم ریخته بازتر می کنم

**** 5 راهنمایی برای اعمال نفوذ طراحی کاربر محور در پروژه متن بازت ****

1. **پیگری قابلیت ها و مسائل مربوط به طراحی**. ایده ها را جمع آوری کن همنطور که پیشنهادات و انتقادات را جمع می کنی. با برنامه یا تغییرات در طراحی به تک تک فید بک ها پاسخ نده. مسائل را برای تولید تم هات بکار ببر.
2. **بدون که برای کی داری طراحی می کنی**. وقتی که داری تصمیمات را برای طراحی می گیری، معمولا باید اولیت گذاری کنی که مخاطبیت کیه؟ کاربرای مبتدی یا حرفه ای.
3. **ساخت پیش نمونه یا نمونه اولیه**. از روش پیش نمونه برای قدرت نفوذت استفاده کن (روی کاغذ - یا کامپیوتر) برای بدست آوردن اون پیشنهاد و انتقادی که نیاز داری.
4. **اینو بدون که برنامه ات از دید یک کاربر چطور کار می کنه**.
5. **از قابلیت ها و تکرار طرح استفاده کن برای ساختن مجموعه خودت: 'hackathons' لیست ایمیل ها و ابزارهای توضیح**.

منابع اضافی

- [10 قابلیت Heuristics] (<http://www.nngroup.com/articles/ten-usability-heuristics>) - **چگونه یک گزارش خوب از مشکلات UX بنویسیم** - استفاده ی UX Methods چی هست و کی از آنها استفاده می کنیم] (<http://www.measuringusability.com/blog/method-when.php>) - **قابلیت روش های تست با Testing Methods** - [استفاده از تجربه کاربران با هزینه کم] (<http://usabilitygeek.com/low-cost-user-experience-testing>) - دانشنامه تعامل انسان با کامپیوتر

نسخه 2 (<http://www.interaction-design.org/books/hci.html>) # امنیت را خسته کننده کنیم

منبع : [Guardian Project](#)

ما داشتیم به این فکر می کردیم که در که در پنج سال آینده چکار خواهیم کرد. یک ایده ی امنیت خیلی ساده و بدون روزنه ' این موضوع خسته کننده در بحث ها در یک سری سطوح آمد. این ما را راهنمایی کرد که از اختراعات و نوآوری های گذشته برای تامین امنیت بدون اینکه بخواهیم بهش فکر کنیم الهام بگیریم. این دیگه در مورد تکنولوژی های جاسوسی برای جیمز باند نیست ' این در مورد تاثیر خیلی کم استفاده روزانه از تکنولوژی موبایل همزمان با ارائه بهترین حد ممکن از محافظت در اطلاعات و ارتباطات روزمره ما . پس اینجا ' به عنوان چراغی هدایتگر و الهام بخش لیست ابتکارت امنیتی مان در گذشته را که تلاش داشتیم کسل کننده کننده باشه را گذاشتیم.

IMAGE - Old-seatbelt-image

خسته کننده [کمربند ایمنی] (<http://blog.esurance.com/seat-belt-history>) (دست و پنجه نرم کردن)

IMAGE -

StateLibQld_1_212036_Cream_pasteurising_and_cooling_coils_at_Murgon_Butter_Factory_1939

خسته کننده مثل [شیر پاستوریزه] (<http://en.wikipedia.org/wiki/Pasteurization>)

IMAGE - versagate3

خسته کننده مثل [گارد محافظ بچه](#) (بچه ها و سگ ها ارزش متنفرند ولی اونا واقعا یک چیز ضروری برای تو خونه هستند)

IMAGE - smokealarm

خسته کننده مثل [دستگاه هشدار دهنده دود](#) (هر چند دوستان ما در [Birdi](#) تلاش می کنند جذابشون کنند)

عکس - childlids

خسته کننده مثل [درب قوطی که کودکان نمی توانند باز کنند](#) (و دوباره اینم از اون چیزهایی هست که پدر و مادر ها را واقعا کمک می کنه ' به بیانی میشه گفت مثل یک نجات دهنده می مونه)

IMAGE - sft_08_01

خسته کننده مثل [ترمز ضد قفل](#) (ماشین های جدید نمونه خوبی از یک سیستم امنیتی می مونن!) عکس - bg_SurgeProtectorBuyingGuide_hero_image

خسته کننده مثل [محافظ برق] (

http://www.lowes.com/cd_Surge+Protector+Buying+Guide_754129873_Learn) (آیا تو ترجیح میدی به جای از دست دادن کامپیوتر 1000 دلاریت ' محافظ برق 2 دلاری از دست بدی)

IMAGE - RestaurantInspectionGradeA

خسته کننده مثل [رستوران](#) (باعث میشه که با استفاده از یک نامه امتیاز معمولی ما با خیال راحت بیرون غذا بخوریم)

IMAGE - dougie_2008_in_his_outwardhound_dog_life_jacket

خسته کننده مثل [حلیقه نجات](#) (حتی سگها هم دوستش دارند)

ما امید داریم به با کارامون روی نقاط حساس ' آسیب پذیر و پیشرفت برای حریم خصوصی خیال شما را راحت کنیم. بهترین نوع امنیت اونی هست که شما احساس نگرانی نکنی تا وقتی که واقعا نیاز باشه (و بعد می فهمی که چجوری بدون اون زندگی کردی)

اینها آخرین ها بود از "Bore-Sec" در سال 2015. سال نو مبارک # طراحی و کاربرپذیری ## درباره این بخش طراحی و کاربرپذیری دو مولفه اصلی هستند که نادیده گرفته می شوند. در قسمت کوتاه ما به شما یک دیدگاه از بالا ارائه می دیم که چطور طراحی کاربر محور

کارایی را بهتر کرده

مقاله ها - مقدمه ای برای اصول طراحی آندروید - امنیت را خسته کننده کنیم - آیا واقعا طراحی بطور متن باز (open source) امکان پذیره ؟ - تفکر طراحی و همکاران جامعه زبان باز (open source) کجا می توانند به هم بیوندن ؟ - 5 راه حل برای اعمال نفوذ طراحی کاربر محور (user-centered design) در پروژه متن باز (open source) - پنج قدم تا استفاده از طراحی در پروژه متن باز (open source) خودتان # تفکر طراحی و همکاران جامعه زبان باز (open source) کجا می توانند به هم بیوندن ؟

منبع : [OpenSource.com] (<http://opensource.com/business/09/10/where-design-thinking-and-open-source-community-collaboration-meet>)

In comparing traits associated with design thinking collaboration and collaboration in the open source community, there are many parallels: open exchange, broad participation, rapid prototyping.

There's also one really interesting contrast: The mindset you tend to see when generating and choosing ideas. But what I'll suggest here is that when you apply the best elements of these two mindsets at just the right time in their respective processes, the results can be pretty amazing. **من با فکر کردن به طراحی شروع می کنم**.

The brainstorming phase of design thinking is probably the most recognizable part of the process. These brainstorming sessions are designed to be open. Lots of ideas are generated, ideas are built on ideas--there are no bad ones. To do that, we have to establish ground rules. [Stanford's d.school posts their list](#), and our rules at Red Hat have many similarities. Rather than criticizing someone's ideas, you have to build on them and suggest your own. No devil's advocates allowed. In this environment, a good collaborator is one who generates and encourages the most ideas from others.

There are a few good reasons for the rules. First, criticizing an idea too soon can kill that idea in its most fragile state. It doesn't give others a chance to carry it forward and improve on it. Second, even the wildest--sometimes worst--ideas could spark a great idea in someone else. This happens all the time. Often the worst ideas and best ideas are closer than they seem. And third, it's essential to create an open, positive environment so people are willing to share their ideas and explore. You've got to believe the best idea is always right on the verge of discovery.

.Community collaboration is similarly open, but can be somewhat less nurturing

A solution is found to have merit when it can stand up to scrutiny. In other words, what doesn't kill your idea makes it stronger.

Here a good collaborator is often the one who finds the most bugs. So rather than looking for the positive aspects to build upon we look for holes. And that's a good thing. As the classic open source saying goes, "With many eyes, all bugs are shallow." It's better to find them in beta. But that doesn't make it easy. In fact, sometimes it can feel like a middle school dodgeball game where you put yourself up against the wall and invite your classmates to take their best shot. There are good reasons why this environment works this way, too. If a solution is going to fail, it needs to fail early so it can fail forward. A tough testing environment isn't going to get easier in the real world.

Second, in open source the code is shared, so there's always a possibility, and even an expectation, for improvement. A solution can always be better. If you're lucky, you'll find lots of people to tell you so. If you're really lucky, they'll offer to help fix it. That's the beauty of the

.model

In both cases, each environment is dependent on an open exchange. Everyone has access to information. Many people participate. Even design thinking processes have a prototype stage. But even though their mindsets may seem in conflict with each other--I believe they're two important functions of the same process, simply applied at different stages. When you're in the early stages of brainstorming and generating ideas--take care to protect creative thinking and the ideas they generate so those ideas can multiply. And when you've got that idea you think is good--then it's time to put it to the test. Find the holes. Invite smart people to do the same. I'll be the first to admit this is far easier said than done. Sometimes it can be nearly impossible. But if the idea is still standing at the end, it'll be all the better for it. And then you know you've really got something

برای دنیای پیچیده

چیزی در مورد اینکه چرا ما به امنیت نیاز داریم' و غیره. ایده یک اپلیکیشن ساده است. اما جهانی که ما اون زندگی می کنیم نه. برای اینکه سازنده نرم افزار خوبی باشید' یکی باید موارد زیر را مدیریت کنه :
- قابلیت استفاده و طراحی - کارایی و دستگاه ها - امنیت و زیر ساخت
(... TBD) #المان های تجربه کاربران موبایل

منبع : [Smashing Magazine](http://www.smashingmagazine.com)

مقدمه تعداد کاربران موبایل و استفاده از موبایل در حال افزایش هست. با کاربران بیشتر که در کارهای بیشتری با موبایل انجام می دهند. ([http://www.businessinsider.com/the-future-of-](http://www.businessinsider.com/the-future-of-mobile-deck-2012-3?op=1)), کانون توجه به این هست که چگونه المان متفاوت را بهتر کنیم تا با هم یک کاربر موبایل را بسازیم.

تجربه کاربری موبایل شامل برداشت و احساس کاربران می شه در قبل ' در حال و بعد از تعامل با موبایل. - از طریق مرورگر و یا یک اپلیکیشن - این شامل ذنجیره ای از استفاده کنندگان موبایل های خیلی معمولی با قابلیت های خیلی کم میشه تا تبلت های با قابلیت های بسیار عالی.

ساختن تجربه های کاربری موبایل که کاربران احساس خوبی بکنند با تحت فشار قرار دادن ما برای فکر کردن به اینکه تا کنون ما چه چیزی را از طراحی کامپیوتر بدست آوردیم. این تشریح کردن تجربه کاربری موبایل به مولفه های کلیدی به ما یک چارچوب مفهومی ارائه می ده که برای ساخت و ارزیابی تجربه های خوب موبایل ' در چارچوب رویکرد کاربرمحور و طراحی برای موبایل. ([http://www.smashingmagazine.com/2011/05/02/a-user-centered-](http://www.smashingmagazine.com/2011/05/02/a-user-centered-approach-to-mobile-design/)), این مولفه ها تجربه کاربری موبایل را شکل می دهند از جمله قابلیت ' چارچوب ' ورودی کاربر ' محتوا و بازاریابی ' به همراه چیزهای دیگر.

المان های تجربه کاربران موبایل * کارکرد * معماری اطلاعات * محتوا * طراحی * ورودی کاربر * چارچوب موبایل * کاربردپذیری * قابلیت اعتماد * انتقادات و پیشنهادات * کمک اجتماعی * بازار یابی

متناسب بودن این المان ها بسته به نوع دستگاه تغییر خواهد کرد(تلفن های مقابل تلفن هوشمند مقابل تبلت) و رابطه ارائه (اپ ها در مقابل وب). این مقاله هر کدام از این المان ها را توضیح می ده و هر کدام را شرح می ده با راهنماهای انتخاب شده.
* کارکرد این امر را ابزارها و امکانات را انجام می دهند که به کاربران این اجازه را تسک را تمام کنند و به هدف هاشون برسند.

دستورات العمل - امکانات اصلی را از کانال های دیگر مربوط به حوزه موبایل در اولویت و ارائه قرار بده . برای یک خط هوایی ' این شامل وضعیت پرواز و چک این پرواز می شه. برای فروشگاه زنجیره ای لوازم بهداشتی سفورا' این شامل پشتیبانی در فروشگاه از طریق خرید

آسان تا بررسی محصولات در موبایل می شه. - ارائه قابلیت های مخصوص موبایل (مثل اسکن کردن بارکد یا شناسایی چهره) ' و افزایش قابلیت با استفاده از ظرفیت های دستگاه های موبایل اونجا که امکان تعامل و لذت کاربران وجود داره. اپلیکشن Old Navy's یک بازی تعجب آور ارائه می ده. - اطمینان حاصل کن که امکانات اساسی و محتوا برای موبایل بهینه شده. برای مثال ' مطمئن شو که پیدا کننده فروشگاه نزدیکترین فروشگاه را بر اساس دستگاه مکان یاب نشون می ده و شماره های تماس می تونند تماس بگیرند. - امکاناتی را اضافه کن به کسب و کار مربوط باشه. برای وب سایت خورده فروشی و اپلیکشن ها' این شامل جستجوی محصول ' وضعیت سفارش و کارت خرید می شه. - ارائه امکانات کلیدی از تمام کانال ها. کاربرانی که به سیستم وارد می شوند باید بتونن تنظیمات شخصی ' صرف نظر از دستگاه یا کانالی که استفاده می شه. اگر یک سری امکانات در موبایل نیست ' سپس کاربران مستقیم به کانالی مثل TripIt که یک شبکه شخصی راه اندازی می کنند وصل بشند. TripIt کاربران را بصورت مستقیم به وب سایتی هدایت می کنه برای راه اندازی یک شبکه. ### خواندنی های اضافی "[اپلیکشن به اشتراک گذاری Bump 3.0 Slashes ' کمتر ثابت میشه بیشتر می تونه باشه] (<http://www.fastcompany.com/1816610/why-entrepreneurs-must-focus-on-saying-no-bump-slashes-features>)", Fast Company

* معماری اطلاعات این امر به انجام تنظیمات قابلیت ها و ارسال مطالب به یک ساختار منطقی تا کاربران را برای پیدا کردن اطلاعات و تمام کردن تسک ها کمک کند. این شامل ' مکان یابی ' جستجو و برچسب گذاری می شود

دستورالعمل - به کاربران لینک هایی به ویژگی های اصلی و محتوا در صفحه اصلی ارائه کن ' اولویت بندی کن بر اساسی نیاز های کاربران. [گالری الگوی طراحی موبایل] (<http://mobiledesignpatterngallery.com/mobile-patterns.php?colid=65438029-72157627607680275>) نمونه های از الگوهای ناوبری اولیه و ثانویه برای موبایل ' خیلی هاشون عمودی هستند به جای افقی مثل وب سایت های کامپیوتر. - به کاربران اجازه بده تا بتونن محتوای اصلی و قابلیت ها را با چند حرکت و یا فشار دادن چند کلید جا به جا کنند. بهینه کردن نوپگیشن برای صفحه های کوچک معمولا گسترده و سطحی هست به جای ژرف. در صورتیکه سه تا کلیک یک شماره جادویی نیست. (http://www.uie.com/articles/three_click_rule) کاربران نیاز دارند که هر تب بشناسند تا بتونند تسک را تمام بکنند. هر لایه اضافی به معنی تب های اضافه و انتظار بیشتر برای باز کردن صفحه ها و پهنای بان بیشتر مصرف شده است. آدرس نوپگیشن نیاز هر دو هم صفحه های لمسی و هم غیر لمسی است. وقتی که برای صفحه های لمسی طراحی می شه مطمئن بشید که سایز نوپگیشن ایتم حتما 30 پیکسل هست. یک کلید میانبر ارائه کن ' که کاربر بتونه با گفتنش یا وارد کردن عددی از (0 تا 9) سریع به لینک دسترسی داشته باشند: - برای کاربران موبایل یک قابلیت اضافه کن ' همنطور که CNN با کلید های دسترسی می کنه ' نه مثل Delta که باید نه تا دکمه فشار بدی. به کاربران یک اشارت جهت نمایی ارائه کن که بدونن کجا هستند و چطور می تونن به عقب برگردند. آدرس پیمایش های موبایل معمولا با این کار را با جایگزین کردن کلید برگشت با برچسبی که به کاربران بگن کجا هستند نشان می دهند. برای وب سایت های موبایل ' از کنوانسیون های استاندارد استفاده کن ' مثل اکنون خانه که به صفحه اصلی بر می گرده ' مخصوصا وقتی نوپگیشن تو همه صفحه نیست. - استفاده مختصر ' واضح ' سازگار و توصیفی از لیل ها برای ایتم های نوپگیشن و لینک ها. در حالی که همواره یک عمل خوبه ' این به یک مساله مهم در دستگاه های کوچک موبایل تبدیل شده.

خواندنی های اضافی - "[فصل 1: مکان یابی] (

<http://mobiledesignpatterngallery.com/mobile-patterns.php?colid=65438029-72157627607680275>)", Mobile Design Pattern Gallery, Theresa Neil

* محتوا به طور دیگر این به مواد در وب سایت شما شناخته می شه (همانطور که Lou

Rosenfeld و Peter Morville آن را در * معماری اطلاعات برای شبکه جهانی وب اشاره کردند
(* محتوا (content) انواع مختلف از مواد در شکل های متفاوت است. مثل متن 'عکس و ویدئو
که به کاربران اطلاعات ارائه می کند.

دستورالعمل - یک ترکیب مناسب و متعادل از محتوا را به کاربران ارائه کن (اطلاعات
محصولات 'محتوای اجتماعی' محتوای آموزشی و پشتیبانی 'محتوای بازاریابی). - از چند
رسانه ای وقتی استفاده کن که به کاربران پشتیبانی ارائه می در چارچوب موبایل 'اضافه کردن
اعتبار به محتوا و پشتیبانی برای رسیدن به اهداف وب سایت. بیشتر وقت ها محتوا چند
رسانه ای بهترین ارائه است وقتی کاربر دنبال یک چیزی برای پرت کردن حواس یا سرگرمی
می گردد (مثل خبر یا یک کلیپ های خنده دار) یا وقتی دنبال چیز با ارزش هست (مثل '
چگونگی استفاده از اپلیکیشن یا یک ویژگی جدید) -همیشه به کاربران اجازه کنترل بر
محتوای چند رسانه را بده با شروع نشدن خودکار ویدئو یا صدا' با اجازه دادن بهشون برای رد
کردن یا متوقف کردن محتوای چند رسانه با اینکه آگاهشون کنی که محتوا چقدر از پهنای باند
را استفاده می کنه. - مطمئن شو که محتوات برای موبایل مناسب است. همانطور که ما
راهنمای قطعه بندب داشتیم وقتی که از چاپی به وب رفتیم' نوشته ها باید کوتاه تر نوشته بشه
برای موبایل ها. عکس ها و مدیاها را برای دستگاه ها بهینه کن; این به این معنی می مونه
سایز کوچکتر برای دستگاه های کوچکتر و مطمئن شو که عکس ها به اندازه کافی شارپ
هستند برای آی پد های جدید. (http://bradfrostweb.com/blog/mobile/hi-res-optimization/). -
مطمئن شو که محتوای اصلیت با فرمتی هست که با دستگاه های مورد نظرت کار می کنه.
حتی الان 'وب سایتی مثل Volkswagen's از کاربران iOS می خواد که فلش را دانلود کنند.
Volkswagen's از کاربرای آی پد می خواد که فلاش پلاگین های پشتیبانی نشده را نصب کنند.
خواندنی های اضافی - "[چارچوب موبایل: اگر بهش شک دارین 'آن را ترک کنید] (Jakob Nielsen, Alertbox - "[Mobile](http://www.useit.com/alertbox/mobile-writing.html)", (http://www.useit.com/alertbox/mobile-writing.html)
Karen McGrane, "Content Strategy Link-o-Rama 2011."

طراحی این با نحوه ارائه بصری و تجربه های تعاملی برای موبایل کار می کنه 'شامل
طراحی گرافیکی 'برند و طرح بندی.

دستورالعمل به یاد داری اینو "تجهیز کن 'میناتوریش نکن" از (محبوبیت اثر Barbara
Ballard) و "کوچکش نکن 'دوباره فکر کن" از نوکیا. هر دو نکته اینو را می گه که طراحی
موبایل نباید طراحی تکراری کامپیوترها باشه. - برای قابلیت نگاه اجمالی و اسکن سریع طراحی
کن . قابلیت نگاه اجمالی به این اشاره داره که چقدر کوتاه و ساده میشه اطلاعات طراحی
بصری را انتقال داد. - انسجام بصریت را با نکات و تجربه های دیگه حفظ کن (موبایل '
اپلیکیشن 'وب 'پرینت' و دنیای واقعی) از طریق استفاده از رنگ 'تایپوگرافی و هویت دادن.
شناسایی کردن آمازون در جای شلوغ هم آسونه حتی اگر اسمش قابل خواندن نباشه. - به
کاربران با ابتدایی ترین و برجسته ترین المان های طراحی تا المان های دیگر کمک کن تا بتونن
تسک خودنشون را تمام کنند. این به گردش بصری شناخته می شه. طراحی خوب با خودش
المان های بصری 'اطلاعات معماری' محتوا و کارایی را به برند انتقال می ده و کاربران را
راهنمایی می کنه. - در پروسه طراحی هر دو جهات افقی و عمودی را در نظر بگیر. دستگاه ها
به سرعت دارند از چند جهتی شدن و تنظیم شدن خودکار برای جهت گیری های فیزیکی
سایپورت می کنند. محل کاربر را در صفحه حفظ کن وقتی که اونها جهت را عوض می کنند.
نشان اضافی یا قابلیت متفاوت را اضافه کن در حالت جدید اگر قابل اجراست. برای مثال اپ
ING به کاربران در حالت عمودی یک سری ویژگی های اضافی را اطلاع می ده.
خواندنی های اضافی - "[نمایش محیطی قابلیت نگاه اجمالی] (http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-113.pdf) (6 MB, PDF),
Matthews, Forlizzi and Rohrbach, UC Berkeley [اصول جهانی برای طراحی 'اصلاح و به

روزرسانی] (http://www.amazon.com/gp/product/1592535879/ref=as_li_ss_tl?)

تجدید نظرهای 'تصمیم گیری های بهتر برای طراحی و آموزش از طریق طراحی William Lidwell, Kritina Holden, Jill Butler (ie=UTF8&tag=strategistnet): 125 راه برای افزایش قابلیت استفاده' درک تاثیرپذیری 'افزایش

ورودی کاربر این در مورد تلاش برای وارد کردن اطلاعات می گه ' چرا که این باید به حداقل برسه برای دستگاه های موبایل و استفاده از دو دست نیاز نباشه.
دستورالعمل - ورودی ها را محدود کن تا مفاد ضروری باشند. یا همانطور که Luke Wroblewski در کتابش *موبایل اول* گفته ' وقتی که به فرم های موبایل می رسم می بایست ' به طرز شدیدی کارآمدش کنیم و تر و تمیز' تر و تمیز' فرم های ثبت نام را حداقل کن و از راه حل های دیگه استفاده کن برای پر کردن فرمت ' مثل کد پستی بجای شهر و استان. نمونه ای که واقعا منو همیشه ناراحت می کنه باز بر می گرده به شرکت Volkswagen's ' فرم ثبت نام برای امتحان رانندگی در نسخه موبایل از نسخه وب طولانی تره. - چیز های پر ارزش را هر جا که ممکنه به نمایش بزار. این می تونه شامل آخرین آیتمی که کاربر انتخاب کرده (مثل فرودگاه یا ایستگاه قطار) یا آیتمی که بیشتر از همه انتخاب شده (مثل تاریخ امروز برای چک کردن وضعیت پرواز). - مکانیزم متفاوتی برای وارد کردن ارائه کن بر اساس قابلیت های ممکن دستگاهت. اپ ها از مکانیزم های متفاوت در دستگاه استفاده می کنند از جمله حرکت 'دوربین' گردش نما و صدا' ولی وب سایت های موبایل تازه شروع به استفاده بعضی از امکانات کردند' مخصوصا مکان جغرافیایی (geolocation) - از مکانیزم ورودی های مناسب استفاده کن . کیبورد لمسی مناسب را نشان بده تا به کاربران از جهت نمایی با کیبورد برای وارد کردن اطلاعات کمک کرده باشی. این را بیاد داشته باش که وارد کردن اطلاعات در یک سری از تلفن که فقط کیبورد های شماره ای دارند خسته کننده تره. برای اپلیکیشن های غیر حساس این امکان را به کاربران بده تا بتونن اطلاعات خودشون را ذخیره کنند و در اپ بمانند' از جمله ایمیل 'اسم' برای اینکه موبایل وسایلات شخصی هستند و با تبلت ها فرق دارند که بین چندین نفر استفاده می شن. - سعی کن تکمیل کردن خودکار 'پیشنهادهای چک کردن غلط املائی و فن آوری پیش بینی ارائه کنی تا زحمت وارد کردن اطلاعات و رخ دادن اشتباه کمتر بشه - با امکان اینکه بتونن خودشون هم اطلاعات را وارد کنن.

خواندنی های اضافی "[اشکال در دستگاه های موبایل: راه حل های مدرن] (<http://www.smashingmagazine.com/2010/03/11/forms-on-mobile-devices-modern-solutions/>), Luke Wroblewski

چارچوب موبایل دستگاه موبایل می تونه هر زمانی ' هر جایی استفاده بشه. چارچوب موبایل در مورد محیط و موقعیت استفاده هست - هر چیزی که می تونه تاثیر بزاره به تعامل بین کاربر و رابط کاربری ' چرا که برای موبایل بخاطر چارچوب این می تونه دائما و سریع تغییر کنه. در حینی که ما بر روی سرگرمی ها' مولتی تسکینگ' حرکت' شرایط کم نوری و اتصال ضعیف تمرکز می کنیم' چیزهای فوق العاده دیگری هم هست - فکر کردن به استفاده از تبلت در یک محیط آرام با یک اینترنت پرسرعت.

"[چارچوب تعامل با موبایل] - <http://www.giantant.com/antenna/2007/06/design-sketch-the-context-of-m.html>," Nadav Savio

دستورالعمل - از امکانات و ظرفیت های دستگاه استفاده کن برای پیشبینی و پشتیبانی کاربران در چارچوب استفاده از دستگاه. اپلیکیشن کتاب آشپزی دیجیتال به کاربران اجازه می ده که در هنگام کار از دستور پخت غذا از صدا برای شنیدن دستورات استفاده کنند - امکانی خیلی خوب برای وقتی که دستانت آغشته به کره هستند! - تطبیق سازی را چارچوبت بر اساس ساعتی که کاربران از اپ استفاده می کنند تغییر بده. مثلا اگر در اپ Navfree GPS بصورت خودکار از روز به شب تعویض می کنه' و نقشه ای را نشان می ده که برای رانندگی در شب مطمئن تر هست. - از مکان یاب برای شناسایی کاربران استفاده کن و بهشون محتویات مناسب نزدیک و تخفیف ها را نشان بده. مثلا موتور جستجوی گوگل برای پیدا کردن یک فیلم

نزدیک ترین سینما را به همراه لینک برای خرید بلیط نشون میده. - قدرت نفوذ داشته باش به اطلاعاتی که ارائه کردند و به ترجیحات و تنظیمات آنها احترام بزار. بعد از انجام یک پرواز از چند پروازی که باید انجام می دادم 'TripIt' به من اینها را نشون داد' از جمله پرواز' گیت برای پرواز بعدیم و همچنین مقدار زمانی که تا پرواز بعدی دارم. اپ United این کار را برای من انجام نداد با توجه به این اون بیشتر از من می دونست. می تونست به من پروازم و پروازهای ارتباطیم و اینکه در دو ساعت مانده به پرواز بعدیم کجا راحت از وقتم استفاده کنم' با توجه به اینکه من عضو کلب United بودم. - بهترین حالت پیش فرض تجربه کاربری را برای دستگاه در نظر بگیر تجربه موبایل بری در صفحه های کوچک و یک چیزی مثل کامپیوتر برای تبلت ها) ولی این امکان را به کاربران برای داشتن امکانات پیشرفته بده. یک بحث گسترده در مورد اینکه چطور ما این را به کاربران ارائه بدیم به تازگی اتفاق افتاد با Jakob Nielsen که میگه یک وب سایت جدا برای موبایل (<http://www.useit.com/alertbox/mobile-vs-full-sites.html>) و Josh Clark بحث میکنه برای طراحی پاسخگو یا responsive design (<http://www.netmagazine.com/opinions/nielsen-wrong-mobile>); و بقیه معتقد هستند که هر دو اینها اشتباه می کنند. (-<http://www.netmagazine.com/opinions/nielsen-vs-clark-theyre-both-wrong>)

خواندنی های اضافی - "[چهار چوب تعامل موبایل]" (http://www.giantant.com/output/mobile_context_model.pdf) , (0.2 MB, PDF) Nadav Savio and Jared Braiterman - "[در چارچوب موبایل]" (<http://blog.cloudfour.com/on-mobile-context/>) , Jason Grigsby لینک های منابع در مورد چارچوب موبایل نزدیک به پایان از دست نده. - "[مردم کی و کجا از دستگاه های موبایل استفاده می کنند؟]" (<http://www.lukew.com/ff/entry.asp?1263>) , Luke Wroblewski ### کاربرپذیری این یک اندازه گیری کلی هست برای اینکه چقدر خوب معماری اطلاعات' طراحی' محتوا و دیگر المان ها با هم کار می کنند تا کاربران توانایی به انجام رساندن اهدافشون را داشته باشند.

دستورالعمل - برای کاربران مشخص کن که چه چیزی را می تونن انتخاب کنن' بهش ضربه بزنند یا جا با جا کنن' مخصوصا در دستگاه دارای قابلیت لمسی. یکی از یافته های بزرگ Nielsen Norman Group's [تحقیقات برای قابلیت های استفاده آید](#) این بود که کاربران نمی دانستند چی قابل دست و ضربه زدن است. مشکل دیگه ابهام برای کشیدن بود: وقتی که ژست کشیدن می تونه همونی باشه که برای کار دیگری در جای دیگه استفاده میشه. اطمینان حاصل کن که امکان تاج کردن بر روی ایتیم براحتی اتفاق می افته مثل لینک ها' ایکن ها و دکمه ها حتما مشخص هستند - برای دستگاه های لمسی' اطمینان حاصل کن که اندازه درسته' به خوبی فضا داره تا از خطا هنگام انتخاب کردن جلوگیری کنی. همچنین قسمت انتخابی را به در صفحه قرار بده; - از معاهدات و پترن های پیروی کن تا از پیچیدگی های یادگیری برای کاربران کم کنی و تجربه موبایل را بیشتر ساده بکنی. اپ های اختصاصی باید یک پلتفرم و استانداردها و راهنماهای خاص را دنبال کنند. مجموعه جامعی از لینک هایی رسمی برای UI و راهنماهای UX در این مقاله موجود هست. "on" [UI Guidelines for Mobile and Tablet Web App Design](#) Breaking the Mobile Web. - اطمینان پیدا کن از قابلیت استفاده در شرایط متغیر' شامل تابش شدید نور به دستگاه' تغییر زاویه دید و جهت' با توجه کردن به المان های طراحی' کنتراست' رنگ' تایپوگرافی و سایز. - به تکنولوژی هایی که به صورت جهانی توسط مخاطبان دستگاه ها ازشون ساپورت نمیشه اتکا نکن' از جمله Java, JavaScript, cookies, Flash, frames, pop-ups و auto-refreshing. وقتی که یک پنجره جدید یا جابجایی از اپ به مرورگر باز میشه به کاربران اطلاع بده تا از overwriting بر روی تب های که باز هست جلوگیری کنه.

خواندنی های اضافی - "[قابلیت های اپلیکیشن های آی پد و وب سایت ها]" (<http://www.nngroup.com/reports/mobile/ipad-ui-guidelines/>) , Nielsen Norman Group [راهنمای طراحی](#)

اب های وب برای موبایل و تبلت, " Max Firtman - "[به روز رسانی قابلیت های موبایل] (Jakob Nielsen, Alertbox ", (<http://www.useit.com/alertbox/mobile-usability.html>)

قابلیت اعتماد این به میزان اعتماد ' اطمینان و راحتی کاربران هنگام کار کردن با وب های موبایل یا اپلیکیشن ها داره. [study by Truste and Harris Interactive 2011](#), حریم خصوصی و امنیت تو نگرانی اصلی کاربران کاربران تلفن های هوشمند است. - **حریم شخصی و امنیت دو نگرانی اصلی کاربران تلفن های هوشمند است** .

دستورالعمل - اطلاعات شخصی کاربران را جمع آوری و استفاده نکن (مثل مکان ' لیست تماس) از دستگاه های موبایل بدون اجازه کاربران. طی چند ماه اول امسال گزارش هایی داشتیم که یک سری اپلیکیشن ها بصورت مخفی اطلاعات کاربران را کپی می کردند. (<http://venturebeat.com/2012/02/22/privacy-policy-apps>) and [users retaliating](#) . - برای کاربران کنترل اطلاعاتی را که با برنامه های موبایل به اشتراک گذاشتن را آسان کن با سوال کردن قبل از اینکه اطلاعاتشون را بگیریم. - شیوه ی کسب و کار خودت را واضح کن (از جمله حریم خصوصی ' امنیت و برگرداندن) و محتوای آنها را ارائه کن (از طریق نشان دادن لینک هایی به حریم خصوصی و سیاست های امنیتی در صفحه ثبت نام) سیاست ها خودش باید قابل دسترسی باشه در قسمت دوم تجربه کاربری موبایل (مثل پایین صفحه یا تب " بیشتر") . اعتبارت را تقویت کن با نمایش نشانه های قابل اعتماد ' مخصوصا وقتی کاربران نیاز به اعتماد دارند با اطلاعات شخصیشون یا اطلاعات مالی . - سیاست هات را به درستی با خلاصه اون یا گزینه ای که امکان میده تمام سیاست ها را از طریق ایمیل دریافت کرد ارائه کن. سیاست های حریم خصوصی و امنیت معمولا خیلی طولانی ' خسته کننده و قانونی هستند در نتیجه کاربرها چشم بسته بهش نگاه می کنند ' ولی تو امکان پرینت را برای کسانی که علاقه مند هستند بزار. - کار کاربرها را خراب نکن با نشان دادن نوشته های قانونی. به جایی بیرشون که قبل از مزاحمت تو بودند. (<http://www.useit.com/alertbox/disrupting-users.html>)

خواندنی های اضافی - **"لایه های خط و مشی طراحی"**, TRUSTe Blog

انتقادات و پیشنهادات (فیدبک) این مربوط به متدهایی برای جلب توجه کاربران و نشان دادن اطلاعات مهم میشه.

دستورالعمل - تعداد هشدارها را برای نمایشگرهای برنامه کم کن و مطمئن شو که فقط هشدارهای مهم و بدرخور نمایش داده میشه. برای خنده ' به Chris Crutchfield's نگاه کن. [video on notification and alert overload](#) . - هشدارها را کوتاه نگهدار ' توضیح بده که چیزی باعث هشدار شده و کاربر چه کاری می تونه انجام بده ' به همراه یک لیبیل خوب. - تذکر باید خیلی کوتاه و حاوی اطلاعات مفید باشه ' نه اینکه مزاحم کاربر بشه با هر کاری که انجام می ده و براحتی بشه آن را رد کرد. - فیدبک ها و تاییده ها را طوری در صفحه نشان بده که مزاحم کاربرها نشی. - اگر اپ شما نشانه ها و استاتوس های هشدار را نشان می ده ' نشانه ها را به آدیت شده نگه دار و اونها را پاک کن فقط وقتی که کاربر از هشدار مطلع شد.

خواندنی های اضافی - **"[اعلان های موبایل]** (http://www.avc.com/a_vc/2011/03/mobile-notifications.html)", Fred Wilson - **"[آینده ی اعلان های موبایل]** True Ventures (<http://www.trueventures.com/2011/03/01/the-future-of-mobile-notifications/>)", Theresa Neil

"فصل هشتم: فیدبک و بهره وری"

کمک این مربوط به گزینه ها ' محصولات و خدماتی موجودی هست که به کاربران کمک می کنه هنگامی که از وب سایت یا اپلیکیشن استفاده می کنند.

دستورالعمل - برای کاربران دسترسی به کمک و گزینه های پشتیبانی را آسان کن. معمولا کاربران در پایین صفحه وب سایت ورژن موبایل و نوار ابزار یا نوار زبانه اپلیکیشن به دنبال کمک هستند - راه های متفاوتی برای ساپورت ارائه کن ' از جمله گزینه های در چارچوب موبایل ' مثل سلف سرویس سوالات متداول ' پشتیبانی زنده با کلیک به تماس و near-real-time و پیام مستقیم در تیوتر. دو کمپانی مالی که به صورت دائم از طریق تیوتر پشتیبانی ارائه می

دهند' کمپانی های American Express و Citibank هستند. - یک توضیح سریع و آموزش کوتاه برای استفاده از برنامه' زمانی که برای بار اول راه اندازی میشه ارائه کن ولی این امکان رد کردن و دیدن دوباره وجود داشته باشه. - وقتی که یک امکان جدید را دارید معرفی می کنید' یک متن کمکی و نکته هایی برای راهنمایی کاربران ارائه کنید. - ویدئوهای برای کمک در زمان مناسب ارائه کن' ولی به کاربران اجازه بده ویدئو را شروع' مکث' متوقف و صدا را کنترل کنند.

خواندنی های اضافی - "فصل 7: دعوتنامه" و فصل 9: کمک' گالری الگوی طراحی موبایل: الگوهای UI برای اپلیکیشن های موبایل' Theresa Neil. فصل هفتم آنلاین هست در بخش 6 - UX. طرح برتر برای کمک به طراحی اپلیکیشن های آیفون, "Catriona Cornett, inspireUX

اجتماعی این به محتوا و ویژگی های ربط داره که یک حس مشارکت اجتماعی را ایجاد می کنه' که تعامل با کاربر را قادر می سازه و این تسهیلات به اشتراک گذاری شبکه های اجتماعی را تاسیس کرد.

دستورالعمل - ایجاد و حضور دائم در شبکه های اجتماعی (مثل یک صفحه فیس بوک) و خدماتی محلی (و به عنوان مثال یک صفحه مثل Google Places و Bing Business Portal and Yahoo Local) این نتیجه های جستجو را پررنگ تر می کنه بر اساس مکان در سرویس های شبکه اجتماعی. به اضافه اسم کسب و کارت این ها را هم اضافه کن' آدرس' تلفن' وب آدرس و ساعات کاری. - حضورت و فعالیت هات را در شبکه اجتماعی و در داخل وب سای تجربه موبایل قرار بده با نشان دادن آخرین فعالیت هات و درخواست برای عضویت آن ها در شبکه های اجتماعی. - امکانات شبکه های اجتماعی به وب سایت اضافه کن تا کاربران راحت تر بتونند با شبکه اجتماعی خودتون در ارتباطش کنند. این می تونه ساده باشه یا استفاده از APIs برای به اشتراک گذاشتن' bookmarking و تگ' لایک و اظهار نظر گذاشتن. - کاربران را دعوت کن که با نام و برند تو تا تولید محتوا کنن در مورد محصولت یا سرویست از طریق موبایل های خودتون و در قالبش یک سری تشویق ها بگیرند. به عنوان مثال 'the burger chain Red Robin' کاربران را دعوت می کنه که عکس بچه های خودتون را در حین خواندن کتاب مدرسه به اشتراک بزارند و در قالبش یک میلکشیک بگیرند. - تخفیف های را عرضه کن که می تونه به اشتراک و گذاشته بشه و همه جا پخش بشه. مثلا American Express به تازگی برای کاربرانی که حسابشون کارتوشون با شبکه های اجتماعی با وب سایت سینک (وصل شده) ا تخفیف ویژه گذاشته. (<https://sync.americanexpress.com/>) - اپلیکیشن های به مشارکت های اجتماعی از طرف کاربران وابسته هستند باید باید طوری به محتوا نگاه کنند که بدرد بخور و در نهایت خودکفا باشه. مثلا برنامه TSA یک ویژگی با کمک کاربر داره که مدت زمان انتظار در بخش امنیتی فرودگاه را اعلام می کند.

خواندنی های اضافی "[راهنمایی صریح برای اضافه کردن ویژگی های اجتماعی به اپلیکیشن موبایل] (<http://www.verious.com/marketplace/social-networking.php>)", ## # بازار یابی این درباره منتهایی است که کاربران آن کشف کردند برای تشویق استفاده مکرر از وب سایت' یا اپلیکیشن.

دستورالعمل - اطمینان حاصل کن از یافپذیری با بهینه سازی جستجو و کشف در موبایل' مثلا آدرس URLs را کوتاه نگه دار. اگر آدرس جدایی جدایی بر وب سایت نسخه موبایل' از این چنین URL می تونی استفاده کنی (m.site.com or mobile.site.com) در نتایج جستجو شده در موبایل اینها را اضافه کن' مثل دسترسی سریع به location-based content (آدرس از محل کنونی) و device-formatted options (مثا کلیک برای تماس) - "پاسخ سریع" کدهای (QR) باید رهبری بهینه موبایل را به صفحه اصلی انجام دهند. بجای روش های سنتی که باید زوم کرد یا بدتر از آن' برای پیدا کردن صفحه اصلی کاربران باید کلی تلاش کنند. به عنوان یک نکته QR codes painted on buildings باید بزرگ و اندازه کافی خوانا باشه تا توسط دستگاه های موبایل به

رسمیت شناخته و کشف رمز بشه. - ایمیل کمپین ها باید به همراه لینکی باشه که پیام برای موبایل ها به راحتی باز بشه (mobile-friendly format) که خودش لینکی داشته باشه که در صفحه مخصوص برای کاربران موبایل نشان بده. یا مثل CVS/pharmacy که کاربران را مستقیم به صفحه ی مخصوص موبایل می بره. - اپلیکیشن را از راه های متفاوت ترویج کن (تلویزیون، روزنامه، فروشگاه تبلیغات) و بهشون انگیزه برای دانلود کردن بده تا از اپلیکیشن استفاده کنن. اگر اپلیکیشن قیمت داره سعی کن کاربران ترغیب کنی به خرید در بازار های شلوغ با یک تخفیف ویژه در زمانی مشخص. یک راه دیگر اینکه اپلیکیشن ترویج کنی از طریق Free App A Day marketplace. - به کاربران به موقع اعلام کن تا برای اپلیکیشن امتیاز بدهند و یا بعد از اینکه از اپلیکیشن استفاده کردند با شبکه های اجتماعی به اشتراک بزارند. ولی براشون این امکان را باز بزار که بتونن این اقدام را بعقب بندازند یا کلاً متوقفش کنند. این تنها به زبان زد شدن اپلیکیشن کمک می کنه، بلکه به تو این امکان را می ده که بدونی کاربران به چه چیز علاقه دارند و به چیز نه. "[کنترل کردن نظرات] (<http://www.smallte.ch/blog/>) (read_en_24001.html) با استفاده از یک تکنولوژی از کاربران خوشحال ترغیب کن که نظرات خودشان را به اشتراک بزنند و کاربرانی که خوشحال نیستند نظرات را ایمیل کنند. ### خواندنی های اضافی - "[بازاریابی برای اپلیکیشن های آی پد] (<http://mobileorchard.com/ipad-app-marketing-case-study-flickpad/>)", Chad Podowski, Mobile Orchard - "هنر معرفی اپلیکیشن: مورد تحقیق", John Casey, Smashing Magazine - "چگونه برای اپلیکیشن بازاریابی کنی", Michael Flarup, Smashing Magazine

نتیجه گیری تجربه کاربری موبایل هنوز در حال توسعه است و فرصت ها برای بهتر شدن ادامه پیدا می کنه تا همه چیز پدیدار بشه. ما یک نمای کلی از المان های اصلی تجربه کاربری موبایل را نشان دادیم، همراه با یک سری راهنمایی ها برای شروع کار با هر کدام از این المان ها. تمرکز کردن بر این المان های متفاوت به ما کمک می کنه که یک مجموعه خوب از تجربه کاربری موبایل را کاربایرمانمون بسازیم. # برنامه سازی در اندروید را شروع کنیم عنوان اصلی - * می خوام برنامه های اندروید را بسازم. از کجا باید شروع کنم ؟ منبع [lifelifehacker](http://lifelifehacker.com/)

< Lifehacker عزیز، < من یک سری تجربیات در برنامه نویسی دارم ولی هیچ وقت برنامه نویسی برای اندروید انجام ندادم. علاقه دارم که شروع کنم ولی مطمئن نیستم به چه چیزی نیاز دارم. نیاز به یاد گرفتن برنامه نویسی ندارم ولی به این نیاز دارم که از کجا باید شروع کرد با اندروید. می تونی کمک کنی ؟

< با احترام، < آرزوی گوسفند برقی

Mr. K. Dick عزیز، حتماً شما می دونید که کد نویسی برای اپلیکیشن فقط یادگرفتن نحوه کد نویسی نیست. اگر هرگز برنامه نویسی یاد نگرفتین می تونین اینجا منابع متفاوتی را که برای کد نویسی هست چک کنید. در هر صورت، خیلی چیزهای دیگر هم هست که شما نیاز دارید باهاش آشنا باشید تا بتونید اپلیکیشن های اندروید درست کنید.

نکته: این به معنای راهنمای جامع برای همه مطالب در مورد برنامه ها و منابع نیست. در حقیقت یک راهنمای تلاش کردیم که موارد نیاز داخلش باشه، یک چیزی مثل کتاب. در هر صورت ما بهتون یک نمای کلی از ابزار متفاوت می دیم که شما می تونید ازش برای پیدا کردن اطلاعات بیشتر استفاده کنید. این ابزار نیاز به یک سری تجربیات داره و اگر هیچ وقت کد نویسی نکردین قبلاً شاید بعد نباشه این لینک پایین را چک کنید. (<http://lifelifehacker.com/the-best-resources-to-learn-to-code-1517844722>) در هر صورت اگر آماده هستین که از تئوری و نحوه کد نوشتن به برنامه ساختن برید، اینجا این کاربه که شما نیاز دارید انجام بدین.

کیت توسعه نرم افزار اندروید (یا SDK)

کیت توسعه نرم افزار اندروید (SDK) در حقیقت مجموعه ای از ابزارهاست که به شما برای ساختن برنامه اندروید کمک می کنه. چیزهای خیلی بیشتری خارج از SDK هست که در موردش

حرف می زنم ولی اینجا بعضی از مهمترین های SDK هست :

Eclipse/Android Studio ###

در واقع دو محیط اصلی برای توسعه برنامه های اندروید وجود دارد. IDE برنامه اصلی برای نوشتن کدها و وصل کردنش به برنامه هست. این بهت می کنه برای منظم و ادیت کردن فایل های برنامه ات 'پکیج هایی که نیاز داری را در کتابخانه ها مدیریت و حمایت می کنه' و می تونه اون را تست کنه چه روی یک دستگاه واقعی و چه روی یک شبیه ساز. پیش فرض IDE برای اندروید Eclipse هست. Eclipse بهت اجازه تغییر فایل های Java و XML را میده و همچنین بخش های متفاوت از برنامه ات را منظم می کنه همراه با خیلی دیگر از کارهای دیگر. نسخه ای که شما از گوگل می گیرید به همراه یک بسته مدیریت هست و به شما اجازه می ده که برنامه را آپدیت کنید و آخرین نسخه برنامه های اندروید را داشته باشید همزمان با انتشار اونها از طریق گوگل.

راه حل دیگر استدیو اندروید هست که به تازگی توسط گوگل ساخته شده. مثل خیلی دیگر از پروژه های گوگل 'استدیو اندروید بخشی از یک بتای طولانی مدته. دلیل طولانی بدون پتا برای استدیو اندروید اینه که Eclipse را بعنوان IDE اول به برنامه سازان بدن. این به معنای آن نیست که برای همه هست. به عنوان مثال اگر برای ساختن از کیت توسعه بومی برای بازی استفاده می کنی (اگر بهش احتیاج داری 'حتما خودت می دونی که بهش نیاز داری) Eclipse اجباریه. در هر صورت استدیو اندروید راه حل خوبی اگر می خوای بپری شروع به آینده و قطعاً علاقه مند با مواجه شدن یک سری مشکلات خواهی شد. مهم نیست که کدام IDE را انتخاب می کنی 'استفاده ازش مثل فتوشاپ می مونه: ممکنه کلی کارای باحال باهات انجام بدی ولی کار کردن باهات را موقع استفاده از ابزار متفاوتش یاد می گیری. در هر صورت اینجا می تونه برات یک شروع خوب باشه برای یاد گرفتن اصول اولیه برنامه نویسی برای اندروید. اینجا یک سری منابع و آموزش های خوب هست که می تونی شروع کنی :

* [Udacity - Developing Android Apps](https://www.udacity.com/course/android-development): این این کلاس آنلاین 8 هفته ای مجانی توسط مهندسين گوگل هست با مجموع از المان ها و آموزش های مستقیم. این دوره کلاس تنها کد ها را برات کپی نمی کنه ولی بهت یک سری اصول مهم و امکاناتی که لازم داری را بهت یاد می ده. * [آموزش توسعه اندروید] (<https://developer.android.com/training/index.html>): بخشی از مستندات گوگل شامل آموزش چگونگی استفاده از ابزار هست. این نوشته ها تو را راهنمایی می کنه جهت برخی از امکانات ابتدایی IDE. اگر تجربه کافی در زمینه توسعه برنامه ها نداری 'این تو را استاد نمی کنه ولی بهت کمک می کنه مه ابزار را بهتر یاد بگیری. * [Vogella](https://vogella.com/tutorials/Android/index.html): این ارزشش را داره که درباره آموزش Vogella در هر قسمت بگیریم. مجموعه ای بزرگ از آموزش ها که خیلی چیزها را کاور می کنه. اگر سوال ابتدایی داری این پایین ممکنه جوابت را نگیری' Vogella را چک کن.

ADB ###

ما قبلاً در مورد ADB صحبت کرده بودیم از [دیدگاه کاربران معمولی] (<http://lifelhack.com/the-easiest-way-to-install-androids-adb-and-fastboot-to-1586992378>), ولی وظیفه اصلی این ابزار در واقع کمک به توسعه هست. به عنوان مثال 'این شامل اندروید SDK هست. شما می تونید ازش برای لود کرین نرم افزار یا تغییرات در دستگاهتون زمانی که به کامپیوتر وصل هست استفاده کنید. اینجا یک سری ابزارهای ابتدای هست.

<http://lifelhack.com/the-most-useful-things-you-can-do-with-adb-and-fastboot-1590337225>

می تونید با ADB استفاده کنید 'ولی اگر دوست دارید بیشتر یاد بگیرید این ها را ببینید : * [ADB Documentation](https://developer.android.com/studio/debug/adbd): این منابع اولیه از گوگل برای اینکه ADB چی هست و چگونه کار می کنه. می تونید چیزهای بیشتری در مورد اینکه توانایی ADB چی هست را اینجا پیدا کنید. *

[Vogella - Using the Android Debug Bridge]

<http://www.vogella.com/tutorials/AndroidCommandLine/article.html>) یک آموزش دیگه از Vogell ' این یکی ابتدایی ترین چیزها را در مورد نحوه کاربرد ADB و بعضی کارهای دیگه را که می تونی انجام بدی بهت می گه. اگر نمی خوای سراغ تمام آموزش های گوگل در مورد اینکه چه چیز نیاز داری بری ' شاید اینجا جای خوبی برای شروع باشه

راهنمای توسعه دهندگان اندروید

ما بهتون یک سری لینک دادیم از منابع اصلی [راهنمای توسعه دهندگان اندروید] (<https://developer.android.com/develop/index.html>) تا حالا این به نشون داده که چقد بدرد بخور هستند. گوگل مجموعه بزرگی از آموزش ها و منابع برای برنامه دادن به اپلیکیشن ها داره که تو می تونی اونها را به عنوان منبع استفاده کنی.

اگر بار اولته که با اندروید برنامه می سازی ' ممکنه یکم عذاب آور باشه اگر بخوای بعضی از این درس ها و راهنما ها را بخونی. آنها طوری گذاشته شدند بگونه ای که هر کدام به دیگری ربطی داره. (به آموزش برنامه سازمان اندروید نگاه کنید). اینجا قسمتی از اون را براتون گذاشتیم که می تونید شروع کنید :

* **Google Services**: ما قبلا در موردش صحبت کردیم. [Google Play Services](#) ' ولی الان اینجا می فهمین که داستان از چه قراره. گوگل به شما ابزار های متفاوتی ارایه می ده که ممکنه شما نیاز به ساختنش داشته باشین ' مثل نقشه ' جهت یاب ' سرویس های پیشبینان گیر ابر و یا سرویس ورود و خیلی چیزهای دیگه. می تونی همه را اینجا ببینی. * **API راهنمای**: گوگل سرویس جدا از API معمولی تنظیم می شه ' که می تونید در موردش اینجا بخونید. این از کد گرفته تا ساختن یک انیمیشن ساده تا خوندان سنسور ها تا اتصال به اینترنت. هزاران اطلاعات دیگه هست که می توانی به قابلیت های برنامه ات اضافه کنی. * **نمونه کد**: بعضی وقت ها کمک این بهت کمک بکنه اگر کس دیگری این کار را قبل از تو انجام داده. این قسمت بهت نمونه کد ها را برای قابلیت های متفاوت نشون می ده. این بهت نشون می ده که چجوری کار می کنه یا فقط در برنامه ات بزارش بدون اینکه بخوای دوباره راه رفته را بری. راهنمای طراحی اندروید

همتای راهنمای توسعه دهندگان اپ ' راهنمای طراحی. گوگل بطور گسترده تمرکز کرده به آموزش اینکه توسعه دهندگان اپ تنها اپی را بسازند که خوب کار کنه بلکه ظاهر خوبی هم داشته. به عنوان نمونه ' این کلی کاره برای شما انجام شده برای پوشش چیزهای ابتدایی مثل دکمه ها ' انیمیشن ساده و چیزهای دیگه.

جایی که اطلاعات بیشتری در این مورد کسب می کنی {راهنمای طراحی اندروید} (<https://developer.android.com/design/index.html>) , چرا که اینها بخش دوم عمده آموزش های رسمی گوگل هستند. این را به ذهن داشته باشین که اینه ممکنه درک بصری خوبی برای طراحی که به ساخت ظاهری برنامه ربط داره نداشته باشن. به زبان دیگه ' اگر می دونید که برنامتون چه شکلی قراره بشه شاید به این نیاز نداشته باشید. اگر می دونید که برنامتون چه شکلی میشه ولی به اندازه کافی در برنامه درست کردن وارد نیستید ' این را ببینید

اینجا لیست بعضی از موارد بدرد بخور برای شروع هست :

* **دستگاه ها**: هدف اندروید فقط موبایل ها نیستند. این قسمت بهت کمک می کنه در جهت یاد گیری اینکه موبایل ها ' تبلت ها ' تلویزیون ها و ساعت ها که چطور باهم ارتباط دارن و چطور میشه برای همه اینها برنامه طراحی کنی. * **طرح ها**: اندروید بر روی یک ساختار اینترفیس ساخته شده. این قسمت بهت یاد می ده که چطور اپلیکیشن ها کار می کنند که تو بتونی یک چارچوب درست کنی که و بعد می تونی طراحی را روی اون انجام بدی. * **Material Design Documentation**: عملا این قسمت الان جداسط ' ولی آخرین نسخه اندروید گوگل که بزودی معرفی میشه طوری هست که قسمتی برای زبان ها داره و اسمش Material Design. اینجاست که می تونی بفهمی که این معنایی داره و چطور در مورد طراحی برنامه ها فکر بکنیم که این

درش جا بشه. همچنین بهت کمک می کنه که اگر در مورد فکر کردن برای ظاهر اپلیکیشن تجربه کافی را نداری.

GitHub/BitBucket

زمانی که تو داری اپلیکیشن را می سازی ' کلی فایل هست که تو نیاز به مدیریت دارند و تو به راهی نیاز داری که بشه تغییرات را دنبال کرد. Git یکی از پروتکل هایی هست که برای مدیریت ورژن های جدید و تغییرات درون برنامه ها استفاده می شه. لزوما یکم از ابزار های معمولی بکاپ پیچیده تره. این آزادی عمل داره که به تو اجازه می ده که شاخه های مختلف از برنامه شما را مدیریت کنه و همچنین جلوی نسخه های قدیمی که مشکل دارند را می گیره.

دو تا از مدیریت های سرویس با Git که خیلی استفاده می شه Github و Bitbucket هستند. هر دو از یک پروتکل اساسی استفاده می کنند و این می تونه بصورت مستقیم و یکپارچه وارد Eclipse و یا Android Studio بشه. BitBucket بهت اجازه می ده که از یک سری فضاها (این را بخون : فضا برای پروژه ها) بطور مجانی استفاده کنی ' در حالی که برای GitHub's مجانی باید اونها را به بازدیدکنندگان نشون بدی مگر اینکه بخوای یکم بیشتر پول بدی. اینجا یک سری از منابع را گذاشتم که بهت کمک می کنه با Git شروع به کار کنی:

* [BitBucket Tutorials](#): Atlassian سازنده ی BitBucket یک سری راهنما داره برای اینکه چطور

با BitBucket کار کنی و یا چگونه اونو وارد پروژه ات کنی. با تجربه شخصی من هر دوی BitBucket و GitHub را نصب کن ' سرویسشون و راهنمایی که ارایه می دن خیلی آسون تره برای کسی که تجربه حرفه ای نداره و می خواد شروع به کار بکنه. * [GitHub Guides](#): GitHub هم یک سری آموزش ها داره برای اینکه چطور سرویس را نصبش کنی که می تونی اینجا ببینشون. بعضی از راهنماها برای نسخه قدیمی تر هستند ولی در کل می تونی اون را نصب و استفاده کنی. * [Vogella Git Tutorial](#): Vogella یک آموزش خوب دیگه داره برای اینکه نحوه کارکرد Git را نشون بده و بهتون کمک کنه در تمام طول پروژه تون.

ساختن برنامه برای آندروید بیشتر از قرار دادن Java در متن و ادیتش هست. اگر یکم تجربه داری با کد نویسی آشنا هستی و تا بحال نتونستی برنامه درست کنی ' خیلی چیزها هست که هنوز نمی دونی. خبر خوب اینه که تو اولین کسی نیستی که وارد این مسیر میشی. فقط یک سری ابزار هست که نیاز داری بدونی و امیدوارم که این راهنما تو را در مسیر درست قرار بده.

با احترام ' Lifehacker # اصول اپلیکیشن

- المان های تجربه کاربران موبایل - ساختن اپلیکیشن ها برای دنیای پیچیده - توسعه در اندروید را شروع کنیم #رمزگزاری تمام بیت ها - آشنایی با امنیت موبایل

این یک توضیح از رمزگزاری برای تمام بیت هاست (

https://docs.google.com/presentation/d/1Jgg405aEvDwkm_f65Z7x3qin_vGqzDNsk_p3cPB4V.presentation (Zc/edit#slide=id.g181a3a731_0335

قصد در مقابل اجرا

قسمت (جلسه) بررسی کلی نگاه اجمالی بر برنامه ها و کتابخانه برنامه سازان پروژه گاردین (30m) مدل های تهدید کننده و داستان های جنگی : گفتگوی آزاد در مورد ریسک ' بیم و نیاز های امنیتی (30m) - رمزگزاری بانک های اطلاعاتی: امن سازی ساختار داده ها در فعالیت ها ' خدمات و ارائه دهندگان محتوا (1hr) - رمزگزاری فایل ها: امن سازی فایل های کوچک تا بزرگ به دلخواه (30m) - شبکه های امن: دفاع کردن در مقابل مرد میانی- خنثی کردن SSL, فیلتر و چیزهای دیگر (30m) دست ها در پیاده سازی زمان برای نمونه کار و اشکال زدایی اپلیکیشن های خودتون با امکانات جدید امنیتی (1.5hr)

رمزگزاری مقدمه ای کوتاه - معرفی خیلی سریع

رمزگزاری چی هست ؟ - متن + الگوریتم + کلید = متن رمزگزاری شده - متقارن یا نامتقارن ' شخصی یا عمومی - قابل پیشبینی نبودن: واقعی یا غیر واقعی وسایل مرسوم برای کد

نویسی : OpenSSL, PGP (GnuPG!), BouncyCastle
 رمزگزاری داخلی اندروید - javax.crypto "BouncyCastle" SSL - HTTPS / TLS / SSL باز - دیسک
 رمزگزاری شده کامل - کلید واژه اندروید (< API 18)
 ### جعبه رمز <https://guardianproject.info/code>
 ### پلتفرم CipherKit عکس را اینجا وارد کن
 ### کتاب خانه برنامه نویسان "CipherKit" CipherKit برای برنامه نویسان اپ در اندروید
 طراحی شده تا برنامه های خودشان را مطمئن بکن برای حفظ حریم خصوصی 'امنیت و
 ناشناس ماندن
 ### SQLite: رمزگزاری پایگاه داده ها SQLiteCipher یک SQLite خارجی هست که
 وظیفش شفاف سازی 256-bit AES برای فایل ها در بانک های اطلاعاتی رمزگزاری شده است.
 صفحه ها رمزگزاری می شنند قبل از اینکه چیزی نوشته بشه در دیسک ها و بعدتر رمزگشایی
 می شنند وقتی که خوانده می شن.
 ### IOCipher: رمزگزاری دیسک های مجازی IOCipher یک دیسک رمزگزاری شده هست
 برای برنامه ها بدون اینکه دستگاه نیاز به روت داشته باشه. این از یک کلون معمولی java.io
 API برای کار کردن با فایل ها استفاده می کنه. فقط دسترسی به رمز عبور و باز کردن صفحه
 های مجازی چیزیه که بین برنامه نویسان هست و تمام فایل ها بصورت رمزگزاری می مونه.
 این بر پایه libsqlfs و SQLiteCipher هست.
 ### NetCipher: رمزگزاری ها شبکه ی اطلاعات و یک پارچه سازی تور NetCipher یک
 بهبود دهنده امنیت شبکه است. اون یک تایید کننده TLS/SSL برای کمک به گواهینامه های
 ضعیف در سیستم هست.
 ### هزار یک پله به عقب برگردیم (تا کشف کنیم که نگرانی ما از کجاست)
 ### پایه مدلسازی تهدید از چی نگرانی ؟ aka Possible Attack Vectors چه نوع اطلاعاتی جمع
 آوری می کنی یا چه سرویسی ارایه می دی که ممکنه وسوسه انگیز باشه و یا در در معرض
 قرار بگیری ؟ آیا برای تو امکان خطری وجود داره که از دستگاه های دیگه بیاد یا حتی از برنامه
 ها 'دسترسی فیزیکی یا استفاده از شبکه ؟
 ### داستان های جنگی ؟ تا به حال شده که کارت 'کاربراتورن یا زندگی مشتری هاتون تحت
 تاثیر بد افزار ها یا مشکلات امنیتی قرار بگیره ؟ * آیا کارت توری هست که به دلیل شغلی که
 درس هستی نیاز به امنیت و پنهان بودن داری ؟ آیا در منطقه ای از جهان هستی که کاربران
 ممکنه تحت حمله قرار بگیرند و یا اینکه نقض حریم خصوصی اتفاق بیافته ؟
 ### ساختار خطر * تحلیل قانونی * ریشه ریایی / جیل بریک * مشکلات OS * بروز رسانی های
 کم یاب * فضاهای قابل جابجایی * خدمات کلود * هدف حملات * به اشتراک گذاشتن دستگاه
 ### بدافزار های در حال تکامل بدافزار های در حال تکامل :
<http://blog.trendmicro.com/trendlabs-security-intelligence/mobile-malware-high-risk-apps-hit-1m-mark>
 ### اطلاعات ذخیره شده در یک متن ساده <http://elifelog.org/book/iphone-gps-cache-data>
 ### Forensic Extraction دستگاه های "Universal Forensic Extraction" به راحتی و خیلی
 سریع می تونن همه اطلاعات را از روی موبایل کپی کنند.
 اگر دستگاهی مثل این تو دست آدم نادرست بیفته ' به راحتی می تونه همه اطلاعات رمزگشایی
 نشده در دستگاه را بدزده
<http://www.cellebrite.com/mobile-forensics>
 ### حمله مرد میانی حمله مرد میانی : <http://thehackernews.com/2013/03/t-mobile-wi-fi-calling-app-vulnerable.html>
 calling-app-vulnerable.html
 ### ظاهر قابل اعتماد
 | شناسه | اسم | توضیحات | | | | 1 | صاحب موبایل باشید | اپراتور اول

موبایل. مطمئن باشید از اینکه به دستگاهتون دسترسی کامل دارید، همچنین امنه و رمز یا کد دارید برای وارد شدن بهش | 2 | توقیف شده / کیفری / خلافکار | یک شخص تحت نظر قانون یا گناهکار که بازداشت یا بازداشت میشه [1]; به تلفن دسترسی دارند. شاید فقط بصورت دستی با زور دسترسی داشته باشند یا با ابزار پیچیده دیگری برای باز کردن تلفن دسترسی داشته باشند. | 3 | اپراتور شبکه موبایل | دسترسی به تماس ها و پیام ها (فرستنده / دریافت کننده / محتویات پیام) و برج های مخابراتی که با اطلاعات سر و کار دارند (مکان های سخت) | 4 | کارفرما' خانواده یا سازمان های حمایتی; | کد / رمز عبور صاحب تلفن را بدانند [1] ولی به داده ها و اطلاعات شبکه دسترسی نداشته باشند; ولی امکان دریافت پیام را دارند | 5 | برنامه های مخرب / تروجان / ایلکشن های کالبد شکافی | به یک سری از اطلاعات صاحب تلفن دسترسی دارند [1] بستگی به میزان اجازه دسترسی و رمزگذاری همون قدر که ضریب امنیتی مهم هست. معمولاً ایلکیشن ها برای دسترسی اطلاعات به اجازه کاربر نیاز دارند. |

استفاده کن؛ - از مرورگرهای HTTPS استفاده کن که اطلاعات را بطور محلی ذخیره نمی کنند؛- از ذخیره کردن رمزهای عبور در مرورگر موبایل خودداری کن || ارتباطات جایگزین (در تقابل با محرومیت از خدمات) | - از VPNs یا برنامه ی تور پروکسی برای پنهان کردن منبع IP و ترافیکت استفاده کن؛ - از برنامه ها و سرویس هایی استفاده کن که فقط با WIFI کار می کنند و اگر سرویس دیتا قطع بود کار نکنند؛ - از برنامه هایی استفاده کن که اجازه می ده از موبایلی به موبایل دیگر اطلاعات را بفرستی |

SQLCipher رمزگزاری بانک داده ها
SQLCipher یک SQLite خارجی هست که وظیفش شفاف سازی 256-bit AES برای فایل ها در بانک های اطلاعاتی رمزگزاری شده است. صفحه ها رمزگزاری می شوند قبل از اینکه چیزی نوشته بشه در دیسک ها و بعدتر رمزگشایی می شوند وقتی که خوانده می شن.
- SQLCipher ردپای خیلی کمی داره و خیلی خوب عمل می کنه برای محافظت کردن از برنامه های بانک اطلاعاتی و همچنین خیلی خوبه برای پیشرفت موبایل - افزایش سریع کارایی به کمی 5-15% در بالا برای رمزگزاری - 100 درصد اطلاعات در بانک اطلاعات رمزگزاری شده با استفاده از شیوه هاب خوب امنیتی (CBC mode, key derivation) زیروکانف (شبکه رابانه ای بدون تنظیمات) و رمزنویسی سطح کاربر ارائه الگوریتم از طریق مرور OpenSSL crypto library.

پلتفرم CipherKit IMAGE
دفاع در عمق حمله کردن ها رل سخت کن با چند لایه امنیتی
اصول حداقل دسترسی در دسترسی به دستگاه نباید به همه برنامه و اطلاعات دسترسی داشته باشه
امنیت اطلاعات حداقل کردن تاثیر دسترسی های بدون اجازه به دستگاه روشن یا خاموش

استراتژی 1. تایید هویت 1. رمزگزاری 1. اعتبار
SQLite vs. SQLCipher
~ `` sjlombardo\$ hex dump -C sqlite.db 00000000 53 51 4c 69 74 65 20 66 6f 72 6d 61 74 20 ~
33 00 |SQLite format 3.| ... 000003c0 65 74 32 74 32 03 43 52 45 41 54 45 20 54 41 42 |
et2t2.CREATE TAB| 000003d0 4c 45 20 74 32 28 61 2c 62 29 24 01 06 17 11 11 |LE t2(a,b)\$
....| ... 000007e0 20 74 68 65 20 73 68 6f 77 15 01 03 01 2f 01 6f | the show.../o| 000007f0
|6e 65 20 66 6f 72 20 74 68 65 20 6d 6f 6e 65 79 |ne for the money
sqlite3 sqlcipher.db sqlite> PRAGMA KEY='test123'; sqlite> CREATE TABLE t1(a,b); \$ ~
sqlite> INSERT INTO t1(a,b) VALUES ('one for the money', 'two for the show'); sqlite> .quit
hex dump -C sqlite.db 00000000 84 d1 36 18 eb b5 82 90 c4 70 0d ee 43 cb 61 87 |.6.?..? \$ ~
p.?C?a| 00000010 91 42 3c cd 55 24 ab c6 c4 1d c6 67 b4 e3 96 bb |B?..?| 00000bf0 8e 99 ee
|?|..28 23 43 ab a4 97 cd 63 42 8a 8e 7c c6 |..?(#C??cB
~ \$ sqlite3 sqlcipher.db sqlite> SELECT * FROM t1
اطلاعاتی نیست ````

https://github.com/sqlcipher/android-database-sqlcipher
وارد کن : net.sqlcipher.database.SQLiteDatabase ; SQLiteDatabase.loadLibs(this)
```` ; SQLiteDatabase db = eventsData.getWritableDatabase("my password")  
## راه حل ساده ما یک بسته SDK ساده را برای برنامه نویسان اندروید درست کردیم تا SQLCipher را در سه مرحله به برنامه هاشون اضافه کنن :  
\* اضافه کردن یک sqlcipher.java و چندتا دیگه. تا فهرست libs برنامه ها \* به روزرسانی مسیر ورود از android.database.sqlite.\* به info.guardianproject.database.sqlite.\* برای هر منبع فایلی

این مرجع است. از android.database.Cursor اصلی بدون تغییر هم می شه استفاده کرد. \* وارد کردن پایگاه داده در onCreate و فرستادن آرگومان متغیر به پایگاه داده باز با رمز عبور\*: -

```

 SQLiteDatabase.loadLibs(this); //first init the db libraries with the context -
 :SQLiteOpenHelper.getWritableDatabase("thisismysecret")
 }

 ## # ویزگی های - Key Derivation - Random salt - Random IVs - AES 256 CBC - SQLCipher
 Fast startup - No size limit - MAC SSL باز -
 ## # چگونه کار می کنه ؟ - پیگری کد ها - Key Derivation - رمزگزاری - MAC
 ## # پیشرفته - PRAGMA kdf_iter - PRAGMA cipher - PRAGMA rekey
 ()cipher_page_size - PRAGMA cipher_use_hmac - ATTACH - sqlcipher_export
 ## # رمزگزاری سامانه فایل های مجازی
 IOCipher یک دیسک مجازی رمزگزاری برای برنامه های اندروید ارایه می ده بدون نیاز دستگاه به وصل بودن جایی. این از یک کپی برابر اصل استاندارد java.io API برای کار کردن با فایل ها استفاده می کنه پس برنامه نویسان از قبل می دونن چجوری باهاش کار کنن. فقط تحویل رمز عبور و باز کردن دیسک های مجازی چیزیه که بین برنامه نویسان و فایل های رمزگزاری ذخیره شده هست. این بر اساس و SQLCipher است.
 IOCipher از فامیل SQLCipher برای آندروید محسوب میشه از اونجا که براساس SQLCipher و از همون روش repurposing یک API استفاده می کنه که برنامه نویسان می دونند. اون در بالای libsqlfs ساخته شده یک فایل سیستمی اجرا شده در SQL که FUSE API را به نمایش می زاره.
 ## # پلتفرم CIPHERKit image
 ## # IOCipher: ویژگی های اصلی * سطح برنامه شفاف و امن دیسک مجازی رمزگزاری
 شده نیاز نداره به جایی وصل باشه سه را جدید برای یاد گرفتن : new
 VirtualFileSystem(dbFile), VirtualFileSystem.mount(password), and
 VirtualFileSystem.unmount() * پشتیبانی کردن از اندروید نسخه 2.1 و پایینتر * مجوز گرفتن با
 +LGPL v3
 IOCipher: The Stack info.guardianproject.iocipher - Java/JNI wrapper API * LibSQLFS / ## #
 FUSE - فایل سیستم مجازی که مسیر می ده به * SQL schema/ structured database
 SQLCipher لایه های رمزگزاری شده برای SQLite * SQLite - اساس مکانیزم ذخیره سازی
 ## # اضافه کردن IOCipher به برنامه - مدیریت کردن رمز عبور - متصل کردن دیسک
 رمزگزاری شده با استفاده از VirtualFileSystem(dbFile) - نصب با استفاده از رمز عبور
 VirtualFileSystem.mount(password) - replace the relevant java.io import statements
 within info.guardianproject.iocipher, e.g.: - import info.guardianproject.iocipher.File; - import
 info.guardianproject.iocipher.FileOutputStream; - import
 info.guardianproject.iocipher.FileReader; - import
 info.guardianproject.iocipher.IOCipherFileChannel; - import
 info.guardianproject.iocipher.VirtualFileSystem; - import java.io.FileNotFoundException; -
 import java.io.IOException; - import java.io.InputStream; - import java.nio.channels.Channels;
 ; - import java.nio.channels.ReadableByteChannel
 ## # نمونه https://github.com/guardianproject/IOCipherExample
 import info.guardianproject.iocipher.File; import ``
 info.guardianproject.iocipher.FileOutputStream; import
 ;info.guardianproject.iocipher.VirtualFileSystem
 ;"File dbFile = getDir("vfs", MODE_PRIVATE).getAbsolutePath() + "/myfiles.db
 ;vfs = new VirtualFileSystem(dbFile)

```

```

// TODO از رمز عبور سخت استفاده نکن! سریع رمزت را بزن! vfs.mount("رمز عبور
دورغیم");
File file = new File(dirPath); File[] files = file.listFiles();
CacheWord # مدیریت کلمه رمز امن
CacheWord یک پروژه کتابخانه اندروید برای ذخیره رمزهای عبور و مدیریت. این به اپ سازان
کمک می‌کند تا با امنیت تولید نگه‌داری کنند و به اسرار به دست آمده دسترسی داشته باشند
از طریق رمز عبوری کاربری. 1. مدیریت کردن رازها: 1. ذخیره سازی عبارات عبور: نگه
داشتن رمزهای عبور در حافظه برای جلوگیری از وارد کردن اون بطور مکرر توسط کاربر
CipherKit [Image] پلتفرم ###
ویژگی‌های CacheWord مدیریت می‌کند منبع اصلی تأیید کردن مقاومت
ریست رمز عبور و کش کردن موارد مخفی اصلی در حافظه. - سرچشمه اصلی قوی (
PBKDF2) - امنیت فضای مخفی (AES-256 GCM) - هشدار دادن مداوم: به کاربر هشدار بده
که اطلاعات برنامه هاشون باز هست تنظیم مهلت زمانی: بعد از هر بار عدم استفاده خود
برنامه بسته بشه - پاک کردن دستی: کاربر بتونه به اجبار برنامه را ببندد - استفاده از منبع
گواهی‌های اندروید بر x.4 اگر موجود هست - هنوز عملی نشده.
[IMG] مشکلات با اندروید ###
CacheWord [IMG] راه حل ###
CacheWord نمونه کد https://github.com/guardianproject/cacheword/tree/master/sample
public class CacheWordSampleActivity extends Activity implements
;ICacheWordSubscriber { ... mCacheWord = new CacheWordActivityHandler(this)
{ } } Override public void onCacheWordLocked@
Override public void onCacheWordOpened@
key = ((PassphraseSecrets) CacheWordService
{ ;()mCacheWord.getCachedSecrets()).getSecretKey
Override public void onCacheWordUninitialized()@
{ mCacheWord.setCachedSecrets(PassphraseSecrets.initializeSecrets(CacheWordSampleActivi
ty.this, "my secret passphrase")); }
NetCipher شبکه امن ###
CipherKit [image] پلتفرم ###
NetCipher: 3 1 دلیل: سوکت‌های قوی‌تر: از طریق حمایت برای مجموعه‌های صحیح
رمز پین کردن و چیزهای دیگر ما مطمئن می‌کنیم که ارتباطات رمزگذاری شده شما به
بهترین شکل ممکن قوی هستند. 1. پشتیبانی از اتصال پروکسی: HTTP و ارتباط پروکسی
SOCKS پشتیبانی ارائه می‌دهد برای HTTP و HTTP/S traffic از طریق تنظیمات خاص Apache
OrbotHelper 1. HTTPClient library: یک کلاس سودمند برای حمایت از ادغام اپلیکیشن با
Orbot: تور برای اندروید. چک کنید اگر نصب شده کار می‌کند و غیره.
[image] خطرات شبکه پروکسی تور ###
NetCipher نمونه کد https://github.com/guardianproject/NetCipher
OrbotHelper oc = new OrbotHelper(this); if (!oc.isOrbotInstalled())
;oc.promptToInstall(this); else if (!oc.isOrbotRunning()) oc.requestOrbotStart(this)
;StrongHttpsClient httpclient = new StrongHttpsClient(getApplicationContext())
if (pType == null) httpclient.useProxy(false, null, null, -1); else if (pType ==
Proxy.Type.SOCKS) httpclient.useProxy(true, "SOCKS", proxyHost, proxyPort); else if (pType
== Proxy.Type.HTTP) httpclient.useProxy(true, ConnRoutePNames.DEFAULT_PROXY,
;proxyHost, proxyPort)

```

## از اینجا <https://guardianproject.info/contact> IRC (freenode): #guardianproject  
 Guardian-Dev and SQLCipher mailing lists <https://dev.guardianproject.info>  
 support@guardianproject.info # چگونه از اطلاعاتم محافظت کنم (IOCipher دیسک های مجازی کدگذاری شده)  
 منبع: [Guardian Project](#)  
 IOCipher یک دیسک مجازی رمزگذاری برای برنامه های اندروید ارایه می ده بدون نیاز دستگاه به وصل بودن جایی. این از یک کپی برابر اصل استاندارد java.io API برای کار کردن با فایل ها استفاده می کنه ' پس برنامه نویسان از قبل می دونن چجوری باهاش کار کنن. فقط تحویل رمز عبور و باز کردن دیسک های مجازی چیزیه که بین برنامه نویسان و فایل های رمزگذاری ذخیره شده هست. این بر اساس و SQLCipher است. (<http://sqlcipher.net>).  
 IOCipher را میشه گفت فامیل SQLCipher برای اندرویده و همچنین این بر اساس SQLCipher و استفاده اش و کاربردش برای همونطور مثل repurposing API می مونه که برنامه نویسان از قبل می دونن. این در بالای libsqlfs ساخته شده ' که یک اجرای filesystem را در SQL که FUSE API نشان می ده.  
 ## ویژگی ها - سطح برنامه شفاف و امن دیسک مجازی رمزگذاری شده - نیاز نداره به جایی وصل باشه \* سه را جدید برای یاد گرفتن: `VirtualFileSystem.get()`, `VirtualFileSystem.mount(dbFile, password)`, and `VirtualFileSystem.unmount()` - از نسخه 2.1 و یا پایین تر اندروید پشتیبانی می کنه - مجوز گرفتن با LGPL v3+  
 ## اضافه کردن IOCipher برنامتون  
 الان باید این کارها در انجام بدی تا کدات از فضای رمزگذاری شده ی IOCipher برای فضای فایل برنامه هات استفاده کنند:  
 1. مدیریت رمز عبور با استفاده از Cacheword یا 1 whatever. گرفتن VFS singleton با استفاده از `VirtualFileSystem.get()` 1. انبوهی از رمز فایل پایگاه داده ها با استفاده از `VirtualFileSystem.mount(dbFile, password)` 1. replace the relevant java.io import statements with `info.guardianproject.iocipher`, e.g.  
 برای اطلاعات بیشتر این ها ببینید [IOCipherExample](#), [IOCipherThreadTest](#), and [IOCipherTests](#).  
 To start from scratch, follow [the instructions on starting with SQLCipher-for-Android](#), then download IOCipher and add it to the libs/ folder of that new project  
 ## دانلودها اینجا می توانی IOCipher jar و فایل های مادری کتابخانه ای را دریافت کنی 'آماده انداختن در پروژه ات ( برای MIPS نیاز داری که از منبعش بسازی):  
 \* [IOCipher-v0.2.zip](#) \* [detached gpg signature](#) \* MD5: d4fdec3ecfaee96277eaa0372db092ce \*  
 SHA1: f89509b802f2982e62b3e5b31cb90266e95262a0  
 اگر به تجربه اساسی کتابخانه FUSE علاقه دارید' می توانید libsqlfs source tarball اختیاری را دانلود کنید:  
 \* [libsqlfs-1.3.tar.bz2](#) \* [detached GPG signature](#) \* MD5: 867d60bcd1cb19f09b1fd3b7112767ac \*  
 SHA1: 3d5fcd3eef9bf07093987ae98951ab7a4525393a  
 ## فضای منابع کدها \* تنها چیزی که برای پروژه نیاز داری:  
<https://github.com/guardianproject/IOCipher>  
 ## اختیاری: مجموعه تست: <https://github.com/guardianproject/IOCipherTests> یک نمونه ساده برای مدیریت فایل برنامه ها: <https://github.com/guardianproject/IOCipherExample> یک تست خیلی ساده برای برنامه: <https://github.com/guardianproject/IOCipherThreadTest> هسته اصلی: <https://github.com/guardianproject/libsqlfs>

## استفاده از نوشته ها \* فقط از یک mount فعال برای هر برنامه پشتیبانی می شه. \* دسترسی single thread/sequential استفاده از IOCipher را ترجیح می ده. \* امکان دسترسی چند موضوعی (multi-threaded) \* الان VFS برای بهینه سازی عملکرد دارای beginTransaction و completeTransaction هست قسمتی از java.io هنوز ساپورت نمی شه : vectored I/O, memory-mapped files

## گزارش دادن مشکلات لطفا اگر مشکلی یا ایرادی داشتین با این کتابخانه به ما اطلاع بدین ! ما منتظر نظرات شما هستیم. این به ما کمک می کنه که مشکلات و ایرادات این برنامه را رفع کنیم. در ضمن خوشحال می شیم اگر شما ایده دارید یا راه حلی برای بهبود برنامه پیدا کردید با ما در میان بزارید.

[گزارش مشکلات و ایرادات] (<https://dev.guardianproject.info/projects/iocipher/issues/new>)  
 چطور اطلاعات برنامه خود را حفاظت کنیم منبع: (<https://guardianproject.info/code/sqlcipher>)  
 مقدمه در زمانی که حفظ حریم خصوصی در گوشیهای همراه در صدر خبرهاست، این پروژه امن کردن درست اطلاعات ذخیره شده توسط برنامهها را برای برنامه نویسان هر چه آسانتر میکند، و همچنین حریم خصوصی کاربران را نیز حفظ میکند. اطلاعات ذخیره شده توسط این نوع از رمزگذاری در برنامه های اندرویدی خصوصیهایی از قبیل: دسترسی کمتر بوسیله نرم افزار های متفرقه، حفاظت اطلاعات در برابر گم شدن یا دزدیده شدن گوشی، و در برابر برنامه های شنود که روزانه بیشتر و بیشتر از آنها برای کپی کردن داده های گوشیهای همراه در گذرگاههای مختلف استفاده میشود، را دارند.

سایفر اس کیو ال (<http://sqlite.org/>) افزونه ایست که قابلیت رمزگذاری با استاندارد bit-256 AES را به بانکهای اطلاعاتی میدهد. تا این زمان این افزونه با حمایت مالی و پشتیبانی (<http://zetetic.net/>) بصورت کد منبع باز میباشد. در دنیای گوشیهای همراه، سایفر اس کیو ال با استقبال زیادی در گوشی های شرکت اپل (<http://sqlcipher.net/documentation/ios.html>) و نوکیا ([http://www.qtcentre.org/wiki/index.php?title=Building\\_QSQLITE\\_driver\\_with\\_AES-256\\_encryption\\_support](http://www.qtcentre.org/wiki/index.php?title=Building_QSQLITE_driver_with_AES-256_encryption_support)) مواجه شده. ناگفته نماند که اندروید بصورت پیش فرض از بانک اطلاعاتی اس کیو ال لایت پشتیبانی میکند، هدف ما ایجاد رابط برنامه نویسی نرم افزاری مشابه برای اس کیو ال سایفر است بطوری که هر برنامه نویسی با هر تجربه و دانشی بتواند با صرف زمانی کم برای یادگیری از آن استفاده کند.

دانلود کنید و بیشتر بخوانید: <http://sqlcipher.net/open-source> منبع: (<https://github.com/sqlcipher/android-database-sqlcipher>)

یک مثال ساده یک بانک اطلاعاتی اس کیو ال لایت بصورت ساده وبدون رمزگذاری است، حتی میتوان کدهای کامپایل شده توسط نرم افزار را مشاهده کرد و تمیز داد. مثال زیر فرق کد کامپایل شده توسط بانک اطلاعاتی اس کیو ال لایت ساده و اس کیو ال با افزونه اس کیو ال سایفر را نشان خواهد داد.

```
hexdump -C sqlite.db 00000000 53 51 4c 69 74 65 20 66 6f 72 6d 61 74 20 33 00 | $
SQLite format 3. | ... 000003c0 65 74 32 74 32 03 43 52 45 41 54 45 20 54 41 42 |
et2t2.CREATE TAB| 000003d0 4c 45 20 74 32 28 61 2c 62 29 24 01 06 17 11 11 |LE t2(a,b)$
.....| ... 000007e0 20 74 68 65 20 73 68 6f 77 15 01 03 01 2f 01 6f | the show.../o| 000007f0
|6e 65 20 66 6f 72 20 74 68 65 20 6d 6f 6e 65 79 |ne for the money
sqlite3 sqlcipher.db sqlite> PRAGMA KEY='test123'; sqlite> CREATE TABLE t1(a,b); $ ~
sqlite> INSERT INTO t1(a,b) VALUES ('one for the money', 'two for the show'); sqlite> .quit
hexdump -C sqlite.db 00000000 84 d1 36 18 eb b5 82 90 c4 70 0d ee 43 cb 61 87 |.6?...? $ ~
p.?C?a.| 00000010 91 42 3c cd 55 24 ab c6 c4 1d c6 67 b4 e3 96 bb |B?...?| 00000bf0 8e 99 ee
|?|..28 23 43 ab a4 97 cd 63 42 8a 8e 7c c6 |..?(#C???.cB
sqlite3 sqlcipher.db sqlite> SELECT * FROM t1; Error: file is encrypted or is not a $ ~
```

``` database

مثال از (<http://sqlcipher.net/design>)

جزئیات برای برنامه نویس ها ما کیت توسعه نرم افزاری برای برنامه نویسان اندروید را آماده کردیم که میتوانند اس کیو ال سایفر را به برنامه های خود در سه مرحله اضافه کنند:

1. Sqlcipher.jar و چندین فایل so. را به دایرکتوری libs برنامه اضافه کنید. 2. آدرس ایمپوت (Import Path) را از `android.database.sqlite.*` به `info.guardianproject.database.sqlite.*` در تمام فایل های منبعی که به آن اشاره شده غیر از `android.database.Cursor` عوض کنید. 3. بانک اطلاعاتی را در `onCreate()` مقدار دهی اولیه (Init) کنید و با رمز `SQLiteDatabase.loadLibs(this);` اول کتابخانه بانک اطلاعاتی را با داده ها مقدار دهی کنید. - `SQLiteOpenHelper.getWritableDatabase("thisismysecret")`

توجه: ما در حال کار بروی اضافه کردن متدهای راهنما (Dialog Builder Helper Methods) برای رمز عبور و شماره رمز (PIN)، بارگزاری رمز عبور (password caching)، و بقیه قابلیت ها که مایلیم در اس کیو ال سایفر استاندارد سازی کنیم.

سازگاری پیش نسخه برنامه نویسان دارای اس کیو ال سایفر V1 است، که با اندروید 2.2 و 2.3 سازگاری دارد، و همچنین فقط با یک پردازنده کار میکند (شما نمیتوانید مقدار Cursor را از سرویس دور {remote service} به activity بدهید).

Notepad + SQLCipher = Notepadbot
Notepadbot یک نمونه اپ هست که از یک نمونه های کد استاندارد اندروید گرفته شده و بروز شده برای استفاده از SQLCipher. می توانید منبع را اینجا جستجو کنید (<https://github.com/guardianproject/notepadbot>) and download the [apk here](#).
سخن آخر خاطر نشان میکنیم که این پروژه قصد جدایی یا دور شدن از پروژه اصلی اس کیو ال سایفر ندارد. ما همکاری تنگاتنگی با تیم اس کیو ال سایفر داریم (<http://zetetic.net>) و قصد داریم این همکاری را ادامه داده و تغییرات و پچ های ارائه شده در آینده را در همین پروژه (کتاب خانه های ساپورت اندروید تحت مجوز [Apache 2.0] هست) (<https://github.com/sjlombardo/sqlcipher/tree/v2beta>) بروز رسانی کنیم.

یک خط همراه با کد Android OS. کد SQLCipher خودش تحت مجوز [BSD-style license from Zetetic LLC] هست. (<https://github.com/guardianproject/android-database-sqlcipher/blob/master/LICENSE>) در یک `original SQLite` و در آخر کد `sqlcipher/blob/master/SQLCIPHER_LICENSE` در یک دومین عمومی هست (<http://www.sqlite.org/copyright.html>). # چگونه از اپ هام محافظت کنم

مقدمه foo

محافظت کردن از دیتابیس اپلیکیشن ها

محافظت و مخفی کردن فایل های چند رسانه ای و اسناد

مقابله در برابر حمله ها علیه شبکه

CipherKit: ما 3 برنامه برای برنامه سازان اندروید طراحی کردیم تا از رمزگزاری بهتر و ناشناس ماندن مطمئن بشوند.

SQLCipher: رمزگزاری پایگاه داده ها SQLCipher یک SQLite خارجی هست که وظیفش شفاف سازی bit AES-256 برای فایل ها در بانک های اطلاعاتی رمزگزاری شده است. صفحه ها رمزگزاری می شنند قبل از اینکه چیزی نوشته بشه در دیسک ها و بعدتر رمزگشایی می شنند وقتی که خوانده می شن. کد SQLCipher GitHub

IOCipher: رمزگزاری دیسک های مجازی IOCipher یک دیسک رمزگزاری شده هست برای برنامه ها بدون اینکه دستگاه نیاز به روت داشته باشه. این از یک کلون معمولی java.io API برای کار کردن با فایل ها استفاده می کنه. فقط دسترسی به رمز عبور و باز کردن صفحه های مجازی چیزیه که بین برنامه نویسان هست و تمام فایل ها بصورت رمزگزاری می مونه. این بر پایه `libsqfls` و `SQLCipher` هست. منبع کد IOCipher

NetCipher: رمزگزاری ها شبکه ی اطلاعات و یک پارچه سازی تور NetCipher یک ارتقاء دهنده امنیت شبکه است. اون یک تایید کننده TLS/SSL برای کمک به گواهینامه های ضعیف در سیستم هست. اون از اجرای پشتیبانی SOCKS و HTTP proxies به داخل برنامه های حمایت می کنه و همچنین از ریشه های پیاز گونه برای ناشناس ماندن و دور زدن ترافیک پشتیبانی می کنه. # چطوری ترافیک اینترنت را امن کنم؟ (NetCipher)

منبع: [Guardian Project](https://guardianproject.org/)

TLS بهتر و یکپارچه سازی برنامه های تور
این یک کتابخانه اندروید که ابزار های متفاوت را برای بهبود امنیت شبکه در برنامه های موبایل ارائه می دهد. اسم پیاز تنها از ریشه های پیاز گرفته شده که تور از این ایده برای گمنام موندن و مقاومت در برابر نظارت ترافیک استفاده کرده بلکه از ایده چند لایه بودن امنیتی برای برنامه ها استفاده می کنه.

جزئیات بیشتری که این کتابخانه می تواند ارائه کند:

1. سوکت قویتر: با ساپورت صحیح از 'cipher suites' پین کردن و بیشتر 'ما تلاش می کنیم تا ارتباطات رمزگزاری شده شما بسیار قویترین شکل ممکن باشه. 1. ساپورت ارتباطات پروکسی: ساپورت از HTTP و SOCKS پروکسی برای HTTP and SOCKS از طریق پیکربندی خاص برای OrbotHelper 1. Apache HTTPClient library: یک ابزار سودمند برای حمایت از یکپارچه سازی با Orbot: تور برای اندروید. چک کنید اگر نصب شده و کار می کنه.
لازم به ذکر است شما می تونید از این کتابخانه بدون Orbot/Tor استفاده کنید ولی ما فکر می کنیم که اگر از ارتباطات TLS/SSL قوی با تور استفاده کنید بهترین چیز تو دنیا می تونه بشه.

این کتابخانه رسماً OnionKit نام گرفته بود - <https://github.com/guardianproject/onionkit>

سوکت قویتر

برنامه سازان می توندن خودشون CACert store درست کنن با استفاده از اطلاعات ما در پروژه

CACertMan : <https://github.com/guardianproject/cacert>

این می تونه در ترکیبی از Android Pinning و MemorizingTrustManager استفاده بشه ' برای ساپورت کردن کاربران نادیده گرفته شده برای گواهی های بدون اعتبار.

ارتباطات پروکسی

وقتی که Orbot با موفقیت به شبکه تور وصل شد ' دو سرور پروکسی ارائه می ده که روی هاست های محلی فعال هستند و برنامه ها می توندن ترافیک خودشون اینجوری منتقل کنند.

HTTP Proxy: localhost:8118 SOCKS 4/5 Proxy: localhost:9050

ان نمونه پروژه اصول استفاده از این کتابخانه را برای باز کردن سوکت ها و ساختن درخواست HTTP از طریق ساکس (SOCKS) نشان می ده و HTTP پروکسی های که موجود را از Orbot. کتابخانه های استاندارد Apache HTTPClient فراخوان هایی ارائه می ده برای ساختن پروکسی. این نمونه کد این را نشان میده. تمام اپلیکیشن هایی که از SOCKS proxy استفاده می کنند نباید DNS را محلی بر طرف کنند ' و بجای اون باید hostnames را از طریق SOCKS proxy عبور بدهند.

کمک رسان Orbot بهتره یک کمک رسان ساده ارائه می ده اگر Orbot (یا تور برای

آندروید) نصب کردید و به چه در حاضر اجرا باشه یا نه. به اپلیکیشن اجازه درخواست شروع Orbot را می ده (چه کاربر اون را فعال کرده باشه یا نه). در آخر این می تونه خیلی سریع به Orbot را نصب کنید ' چه از طریق Google Play, چه بطور مستقیم از APK. داندلود کنید از این سایت ها: torproject.org یا the.guardianproject.info

برای اپلیکیشن هایی با سرور روی دستگاه ' این همچنین می تواند کمک کند به درخواست Tor Hidden Service از Orbot ' و پیدا کردن آدرس اختصاصی ONION.

امنیت و فصل حریم خصوصی

-رمزگزاری تمام بیت ها' آشنایی با امنیت موبایل - چطور از اطلاعاتم و اپلیکیشن هام محافظت کنم - چگونه از اطلاعات رمزگزاری شده محافظت کنم (مقدمه ای برای IOCipher دیسک های مجازی کدگزاری شده) - چطور از اطلاعات برنامه خود را حفاظت کنیم (مقدمه برای sqlCipher دیتابیس رمزگزاری شده موبایل) - چطوری می تونم ترافیک اینترنت خودم رو امن کنم (مقدمه برای NetCipher) - گرفتن عکس های زیرک # گرفتن عکس های زیرکانه

منبع: [Guardian Project](#)

Obscuracam یک اپلیکیشن برای عکاسی و ویدئو برای آندروید هست که یک سری اطلاعات را مخفی نگه می داره.

تا حالا شده از کسی عکس و یا فیلم بگیرین و بعدتر بفهمید که اونها نمی خواستند توی اون باشند؟ راحت نیستید با این که عکس دوستاتون' اعضای خانوادتون یا صورت بچه در اینترنت باشه؟ نگران این هستید که اطلاعات جغرافیایی در عکس مخفی گاه شما را لو بده؟ از این تکنولوژی شناسایی خودکار صورت در فیس بوک' گوگل و دیگر سایت ها خسته شدید؟ پس این برای شماست' این بهتون این قدرت را می ده که از هویت عکس هاتون بهتر مراقبت کنید قبل از اینکه آن ها را به روی اینترنت بزارید.

عکس را بگیرید و یا عکس ها و فیلم هاتون را از گالری وارد کنید' ObscuraCam بصورت خودکار چهره ها را انتخاب می کنه و شما می تونید اونها را شطرنجی کنید' تاریکش کنید و یا با شکلک های مسخره و یا مات کردن اون از هویتش محافظت کنید. شما حتی می تونید کل عکس را شطرنجی کنید و فقط شخص مورد نظرتون را قابل دیدن کنید. این اپلیکشن همچنین همه اطلاعات شناسایی را قبیل نقطه جغرافیایی (GPS) و اطلاعات تلفن و مدلس را پاک می کنه. شما می تونید اون عکسی را که روش کار کردید در گالری ذخیره کنید و بعدتر مستقیم به فیس بوک' تویتر و یا جاهای دیگر به اشتراک بگذارید. ما در حال حاضر درایم یک نسخه از ObscuraCam برای حرفه های می سازیم که [InformaCam] نام داره (<https://guardianproject.info/apps/informacam>). کاربران این قابلیت را دارند که عکس ها امن کنن با PGP ' معکوس کردن برای جمع آوری و پخش دیگر اطلاعات متا دیتا.

چگونه آخرین نسخه را بگیریم

شما می تونید اپ را در [Android Market] پیدا کنید (https://market.android.com/details?id=org.witness.sscphase1&feature=search_result)

دسترسی ندارند می تونید فایل [ObscuraCam.APK] را از پوشه ساخته شده عمومی دریافت کنید. (<https://guardianproject.info/builds/Obscura>). با استفاده از این گزینه حتما بیاد داشته باشید که برای به روز رسانی چک کنید چرا که خود اپ بصورت خودکار اپدیت نمیشه.

پس زمینه

ما دوربین های امن دیگری را با همکاری [Witness.org] ساختیم. (<http://witness.org>), سازمانی فعال در آموزش و پیشرو در حمایت از ویدئو های حقوق بشری. این همکاری در روز هک کردن آزاد سال 2010 در یک ویدئو کنفرانس اتفاق افتاد. (<http://blog.witness.org/2010/10/ovc2010-opensubtitles>)

این یک پروژه منبع باز هست که شما می تونید اینجا پروسه را دنبال کنید.

<https://github.com/guardianproject/SecureSmartCam>

برای بروز رسانی در مورد روند پیشرفت پروژه می تونید این بلاگ را چک کنید:

<https://guardianproject.info/tag/obscuracam>

داستان های کاربر اینجا داستان بعضی از کاربرها هست که ما امید به ساپورت کردنشون داریم' همراه با لینک های که می تونید پیشرفت را مشاهده و تراک کنید:

1. یک گروه فعال فیلمی را از مصاحبه با سخنگو را در یک تظاهرات ضبط کردن و می خوان هویت بقیه آدم ها در فیلم مشخص نشه.

ساختن فیلتر ویدیو و پردازشش در حال انجام هست :

<https://github.com/guardianproject/SSCVideoProto>

1. یک گروه مدافع حقوق از تلفن‌شون برای گرفتن کلی عکس بد رفتاری‌ها در روستا استفاده کردند. در راه برگشتن به هتل 'اونها توسط نیروهای امنیتی بازداشت می‌شن و موبایل هاشون هم مصادره می‌شه.

در مورد پایگاه داده‌ی رمزگزاری شده کتاب خانه SQLCipher بیشتر یاد بگیرید :

<https://guardianproject.info/code/sqlcipher> و برنامه ریزی با امکانات رمزنویسی :

<https://github.com/guardianproject/SecureSmartCam/wiki/Cryptographic-features>

1. شبکه اینترنت و موبایل قطع شده. عکس‌های تظاهرات نقض حقوق بشر باید به خارج از کشور برسه. (tor/proxies/bluetooth/wifi mesh)

Xfer یک اپ برای دریافت و ارسال فیلم هست که در شبکه تور کار می‌کنه :

<https://github.com/guardianproject/sscxfer>

1. تلفن من ممکن است به خطر بیافته. من چیزی را که می‌خواستم انتقال بدم را انتقال دادم. می‌خواهم تمام رد‌های تصاویر گرفته شده را حذف کنم.

InTheClear اپ ما هست برای پاک کردن دیتا / و ترس :

<https://github.com/guardianproject/InTheClear>

Building an Open Source Community #

Source: [OpenSource.com](https://opensource.com)

I have [told the story](#) of how FinTP, the first open source application for processing financial transactions, was born. Here, I would like to present a deeper view on how the community is being built, its structure and governance and why I think people should join such communities. I myself am a founding member of two communities, the first being built together with my friends after our first bachelor party, from a desire to preserve our spirit of joy and good vibes.

The second one is the open source community around FinTP, called FINKers United

An open source project cannot succeed without a powerful community that supports its development. At the same time, Eric Raymond said in [The Cathedral and the Bazaar](#) that a necessary pre-condition for success in an open source project is having something runnable and testable to play with. Therefore we have the chicken and egg dilemma of which comes first—the product or the community. Fortunately, FinTP had both, being the result of evolution from its previous commercial version to open source and having a company, partners' network and a user base that would become the starting point for the open community FINKers United

While it may seem that the transition from the closed community to FINKers United is destined to be smooth, there are many adjustments to be made. The roles in the community have to change drastically, from the former way of interaction between supply and demand sides, in order to benefit from all the advantages an open environment has to offer—reduced time-to-market, better results delivered from a strong collaboration, cost savings. Of course, a coherent governance strategy has to be established so the project doesn't go off rails

Community structure # #

Communities are groups of individuals sharing common interests. FINKers United is not different, all members having the same goal to improve the processing of financial transactions based on open source applications, pushing the interoperability level in order to achieve a new semantic standardization for financial transactions. For this specific community and mission, several professional profiles of individuals and institutions are welcomed: technical (developers, IT architects and designers, implementers, support, quality assurance), business (business

knowledge in banking practice, standards, financial transaction processing, banking or corporate treasury, corporate to bank business etc.), communication, marketing, market analysts, auditing and legal consultants.

In order to successfully establish the strategy and objectives, to manage and control the activity of the community, the community we conceived has the following three governing entities: General Assembly, Board of Directors and Censor. In the early stages of the community, the governance is ensured by the founding company, since it is the sole contributor. For now, the community has a Technical Committee coordinated by the company's CTO, which includes 10 project leaders. In time, new hierarchies will emerge based on merit and contributions, so any member can be elected.

Reasons to join the FINKers United community

?The most common question is what's in it for me, *why should I join this community*

The rewards are very different from the traditional closed-source projects, where there is a clear limit. For suppliers, the types of reward-earning contributions are in the job description, for which you get a monthly paycheck. For the demand side, what you pay is what you get, with limited flexibility from both providers and budget owners.

In open source communities, the benefits are different for every member profile, but the essence is that everybody is involved because of having a shared interest and acts as an entrepreneur—having ideas, initiatives and the freedom to act accordingly to own beliefs.

For the *adopters*, the benefits of using and supporting open source software have been ### :debated in length, and they include

The power to influence projects and strategies - Freedom from vendor locking - Benefit from - better, more secure and reliable products - Better TCO and time to market - Reduce development and maintenance cost - Attract and retain development talent - Community based, 'price affordable, service consistent, guaranteed quality' escrow agent

.For *contributors*, the satisfactions are mostly triggered by a sense of accomplishment ###

Material—custom professional services can be offered to adopters who lack certain skills or - need SLAs - **Intellectual**—there's great opportunity to develop the professional profile by taking on the technical challenges open source projects have to offer - **Recognition**—open source communities reward each of their members based on the value of their contributions - **Competitive**—the opportunity to work for state of the art technological projects, with high impact on several industries

Speaking for myself, my motivation for joining FINKers United is that being a part of the team that attempts to revolutionize the financial transactions processing arena is a great challenge and experience, not to mention that succeeding would be truly an accomplishment. Moreover, I believe that creativity is original thinking based on openly sharing knowledge and transparently sharing results.

Why would you join an open source community # پنج راه برای تعامل بیشتر در گروه

منبع: OpenSource.com

اتاق پر بود با تقریباً 30 نفر از مدیران جوامع متفاوت از کشورهای مختلف. ما در [Community Leadership Summit] در پرتلاند 'اروگان شرکت داشتیم'. (<http://www.communityleadershipsummit.com>) من یکی از جلسه ها را در مورد ساختن برنامه ها و راه های دیگر برای تعامل در جوامع مدیریت می کردم.)

<http://communityleadershipforum.com/t/day-2-session-8a-building-programs-and-other-ways->

(to-engage-your-community/309%20)*. در طول این مدت من چند برنامه را که در Opensource.com برای تعامل بهتر استفاده کردیم به اشتراک گذاشتم. ولی این قسمت فقط در مورد ما نبود بلکه یک فرصت بود تا از مدیران جوامع دیگر در متن باز هم بدونیم. (http://opensource.com/business/14/7/4-lessons-trenches-community-management).

ما مدیرانی از Mozilla, Open Source Institute, Puppet Labs, Chef و غیر داشتیم. ما در مورد اینکه چی کار می‌کنه و چی نه صحبت کردیم. چی امتحان شده چی تست شده. و کی ایده ی جدیدی داره ؟ با هم به لیستی دست از تکنیک ها دست پیدا کردیم که می‌تونه استفاده بشه برای در تعامل نگه داشتن جامعه متن باز (open source).

1. ** خوش آمد به اعضای جدید گروه**

فرق نمی‌کنه که شخصا یا مجازی 'مطمئن شو که عضو جدید گروه خوش آمد گویی خوبی دریافت می‌کنه. برای شخصی که در رویداد هست یک خوشامد گویی را تعیین کن با کسی که خوب لیخنه می‌زنه و اطلاعاتش در مورد مجموعه ات به اندازه کافی هست.

برای خوشامدگویی مجازی 'یک ایمیل خوشامدید می‌تونه کافی باشه. هرچقدر شخصی باشه بهتره. و این باید از طرف مدیر مجوعه یا شخصی باشه که بعدتر در مجموعه فعالیت داره.

1. ** انجام یک ارزشیابی برای تعین علاقه یا تخصص**

یکی از مدیران مجموعه در مورد این توضیح داد که چطور درک و ارزیابی می‌کنند برای اینکه چطور یک شخص به مجموعه اونها اضافه میشه و چطور می‌خوان نقش داشته باشند. آنها این را یک هفته بعد از ایمیل خوشامد گویی انجام می‌دن تا بتون قدم بعدی برای ارزیابی تعامل در مجموعه را ارزیابی کنند.

تکنیکی مثل این نه تنها به اشخاص و یا درون گروه کمک می‌کنه بدانند "چرا" ولی این به مدیران گروه ها کمک می‌کنه تا بدونند که چکار می‌توانند انجام بدهند تا اعضای جامعه در گروهشان رشد کنند.

1. ** اعضای گروه را برای شرکت در رویداد ها کمک کن**

خیلی از سازمان ها هزینه شرکت کردن اشخاص را در رویداد ها پرداخت می‌کنند. این به اونها کمک می‌کنه برای شخص در جلسه ها باشه و اعضای جامعه با هم صورت در صورت دیدار داشته باشند. روابط شکل گرفته از بودن اعضا در جلسه ها 'کنفرانس ها و غیره به رشد اعضا جامعه کمک می‌کنه.

1. **سیلوها را بشکن**

مدیر یک مجموعه دیگر در این کنفرانس 'تکنیک هایی را برای کمک به شکستن موانع از ورود به مجموعه تا تعامل در مجموعه را به اشتراک گذاشت. بصورت ماهیانه این مدیر جلسات آنلاین را تدارک دیده برای ترویج تعامل در مجموعه. اونها از اعضا می‌پرسند که دوست دارند چه چیزی را یاد بگیرند' به اعضا کمک می‌کنند تا منبع یک مشکل تخصصی را پیدا کنند و بعد مدیر اون را به عنوان تایپیک قرارا میده. مجموعه آنها ارزش خیلی خوبی پیدا کرده از طریق اینگونه جلسات و ضریب تعامل کاربران افزایش پیدا کرده.

همزمان با اینکه این مثال ها برای شکستن سیلوها در سازمان های بزرگ زده شده بود' فکر می‌کنم که این ایده ها شایستگی را برای اجرا در هر نوع گروه های دیگری داشته باشند.

1. ** بگو ممنون**

در خیلی از گروه ها از کلمه متشکرم به اندازه کافی استفاده نمیشه! بگو متشکرم' و دوباره از شخص در گروه تشکر کن. در این قسمت' ما لیستی از نحوه تشکر کردن در گروهت و همچنین اشخاص تهیه کردیم.

- برجسب بفرست - براشون خبرنامه یا بلاگ بفرست در شبکه های اجتماعی یا چیزهای دیگر.
- اشخاص در تگ کن و یا عکس ها در شبکه اجتماعی به پست کن (مطمئن بشو که اجازه داری)
(- تی شرت کلاه و یا کوله پشتی بفرست. - از ویژگی های اونها در "کانون توجه گروه" در خبرنامه های و یا صفحه اول وب سایت بگو. - از همه کسانی که به نحوی همکاری کردن برای

کد و یا مطلب در برخی مراحل تشکر کن (مثلا همزمان با انتشار پروژه ها) - برایشون نامه های دست نویس بفرست. - با گروهت شادی کن و یا حمایتشون کن برای شادی کردن خودشون - بهشون امکانات مالی ارائه که بتونن در رویداد ها شرکت کنن. همزمان با اینکه این بخش فقط یک ساعت بود ولی ایده هایی که ما به اشتراک گذاشتیم می مانند و ادامه پیدا می کنه تا جامعه ما تغییر پیدا کنه و بزرگتر بشه. مدیران گروه ها و دیگران می توانند به بحث های دیگر بپیوندند در [Community Leadership Summit online forum (<http://communityleadershipforum.com/>)] و مثل همیشه از شما خواهش می کنم تا تجربیات خودتون را در کامنت ها با ما به اشتراک بزارید. # How to use IRC (Internet Relay Chat)

Source: [Instructable](#)

This is an Instructable intended as a starter for those who do not yet understand Internet [EDIT] Relay Chat, or IRC. This project is not intended to cover the entire scope of IRC and each individual client's capabilities, but is targeted to the windows user who is new to IRC, and this project aims to help those users to get started with IRC as a superior form of communication over IM clients such as AIM or YIM, which require installing software already proven to be a security-risk. Even if you do not choose to use Chatzilla or Firefox, some information here may be useful to new IRC users with general commands supported by most IRC clients. If you are looking for help with a specific client, see FAQ's related to your client specifically. If you have more wisdom to add, please post it as a new comment. For those new to IRC, please read this project and it's comments anyway to make sure you know the basics. You are a user until made an operator, so don't try to kick ops from public channels or your client will laugh at you. All IRC addresses and/or examples are valid and accessible for the purpose of learning how to use the system, to the best of my knowledge. Screenshots were made from actual logons and were not "doctored" in any way, so you can join any channels, public or hypothetical, displayed in any screenshot here [EDIT/]

Internet Relay Chat is much like your instant messengers, but is devoid of spam or general security risks that other IM services not only allow but are built to accept. Typical IM services such as YIM, MSN-chat, AIM, and many others are specific targets due to their accessibility and ability to silently upload files to your computer and execute them on command without you even knowing what happened, from someone not even showing up in your chat window. IRC is not only secure in multiple ways, but you cannot hide behind an application. If someone's there, you'll know it, and you must accept all file transfers before they are actually received. Below, I'll explain how using IRC is just as easy, far safer, and much more functional than your bloated n00b IM-client-for-dummies. For those smart enough to use Firefox, this can't be easier. For those still using IE for anything, there may be no saving you, but I'll try. Step 1:

Learn to use Firefox, or download a suitable IRC client such as mIRC

Step 1: Learn to use Firefox, or download a suitable IRC client such as mIRC ##

You can find Firefox [here](#) * Get the Chatzilla addon [here](#) * For those so stubborn not to try the * best, you can get mIRC [here](#)

I'll focus on those who know better to use Firefox than IE. Downloading the Chatzilla extension and incorporating it into Firefox is easy, simply download it and it knows where to go. You may be asked to restart Firefox or your computer, if you are not asked, restart your computer just to assure a correct install anyway. Windows likes to corrupt itself and blame properly-coded software.

Now that you have it installed, you can find chatzilla from the top by dropping down the "tools" tab in the menu-bar. Step 2: Join a network

Step 2: Join a network This seems largely difficult, but it is incredibly easy, and your ## choices are laid out for you. In this image, I have joined "efnet" as my primary host and server.

.By now you should have connected to efnet for sake of example IRC addresses work similarly to http "hyperlinks" (aka "links"), so that clicking on a link to an .irc channel will automatically start chatzilla for you. Below, I will get you started Start chatzilla, and you will be faced with a lot of messages. Ignore them for now and type the :following into the text-box at the bottom

networks/

:Chatzilla will return this to you in the message window (or similar)

Available networks are dalnet, efnet, freenode, hispano, hybrid net, ircnet, mozneta, {INFO}

.quake net, serenia, slashnet, undernet, webbneta

The networks will be like hypoeerlinks, so you can just click on one to join that particular server.

Here I have chosen to use "efnet" as my server by simply clicking on the name shown above.

Now that I am on a network, I have the ability to join an existing "room" (the real name is a "channel", so we are calling it a "channel" from now on), or create a new one (it is best to start your own channel than to hijack an existing one). So, let's say you and your friends decide to meet on irc://efnet/instructables. Firefox users can simply type this address into the URL window in Firefox, and chatzilla will automatically connect to that address. If you have it

:installed, try it below by simply clicking on the link

[irc://efnet/instructables irc://efnet/Instructables]

Congratulations, you are now using IRC. The link above is shown just as you would type it into Firefox's URL textbox (the box at the top that shows your current internet address). Remember to type an irc address as shown above. Generally, "http" means "internet", "irc" means "irc", of course, and "file" means a location on your computer. You may start to see how the internet and .your computer work much in the same way

Now, say you don't want to use irc://efnet/Instructables, maybe you want your make your own channel to chat in, and you want to call it "Jedi Knights". You can't make this work properly because "channels" (what some call "chat-rooms") cannot have spaces in the names. The ideal workaround to this is to use an underscore instead of a space, as seen below (go ahead and try :it)

[irc://efnet/Jedi_Knights irc://efnet/Jedi_Knights]

Notice that I still have efnet in the address. This is because efnet is the server that is hosting your traffic (your text, files, etc). By now you might start to see how you can create your own places to chat. You need to use a valid server, but you can make up any name you want for your channel as long as it contains only regular characters such as letters and numbers, no special characters will be recognized. This helps your security because an underscore can be printed by over a hundred different key combinations (underscore is often used to display a "non-display .character")

You don't even have to use efnet, just type /networks to see what networks are available, and your address needs to be prefixed with a valid IRC server as seen when typing the /networks command. You cannot simply create your own server, and doing so will not be covered here.. Click on the name and you can find or create your own channels (or "rooms"). What server you

use amongst those available does not particularly matter except for their performance, and they all are generally reliable. However, you cannot go to `irc://efnet/instructables`, and be in the same place as `irc://dalnet/instructables` because you will be on a different server

...Step 3: Now that you are on IRC ##

Picture of Now that you are on IRC... With IRC, you tell the "client" (chatzilla, in this case) that you are giving it a command. The standard is to precede the command with a slash. As seen previously, typing `/networks` tells the client to display the available networks. To see a list of all available commands, simply type `/commands`. A list will be displayed showing the proper commands that are accepted from where you are. I won't get into them all, but I'll show you the important and most useful ones

Probably the most useful to anyone at first is the `/nick` command. This changes your "screen name" to whatever you want, within a limit of 9 characters. Let's say you started with the nickname "IRCMonkey" as chatzilla defaults to, and you want to be "B_Kenobi". Simply type
:as shown below
nick B_Kenobi/

Now that is your new nickname. By default, chatzilla will save this as your default nickname for that channel with a limit of 9 characters. When you select "open this channel at startup" (selectable as shown in the image displayed below, this becomes a new default startup setting), you simply have to start chatzilla and your username and preferred channel will automatically start and you are ready to go. You will not have to change your nickname every time you start, as chatzilla will retain your last nickname before you close it to be used when you return to that channel

If you are stuck, or just want more information, simply type `/help` and chatzilla will walk you through any command. For example, we will ask chatzilla how to use the `/dcc-send` command
help dcc-send/
[[] dcc-send [USAGE]

Offers a file to [nickname]. On a query view, [nickname] may be omitted to send the [HELP] offer to the query view's user. A file may be specified directly by passing [file] or, if omitted, selected from a browse dialog

(Please note that the above apparent links do not lead to anything and were not intended)
This may seem confusing, so let me explain it. Say you are chatting with your friend "Gozer", and you want to send them a file called "my_pix.bmp" from your desktop. You simply type
dcc-send gozer my_pix.bmp/

Gozer will receive an offer to accept or deny this file, and if they choose to accept it, a new tab will open on your side showing the progress of the file transfer and the rate at which it is being transferred. The file transfer is directly from your computer to theirs, and no other people chatting with you will even know it is taking place, much less the server itself. It is transferred by a process called "P2P", which is a direct connection directly between you and your recipient. It is not encoded, but the security lies in that the transfer cannot be redirected, located, or even observed because it is just between you. The two computers are passing data as if they were connected by a private network, and cannot be intercepted because only two connections will work....The sender and the receiver. There is virtually no way a third party can get on that data-stream because the connection will either block them or self-terminate, as it is not compatible with more than two users. The transfer will continue in the background as you continue

.chatting, so there is no need to wait for it to finish
There is a variant of the /dcc protocol, which is /dcc-chat. This allows you to bypass the server and the two of you act as your own. in a /dcc-chat enabled conversation, you are literally talking just between your two machines on a direct line. Locating this type of chat protocol is nearly impossible unless someone is specifically monitoring your line. to enable this with your friend

:Gozer, just type
dcc-chat gozer/

A new tab will open, and you two are literally chatting client-to-client, without a server as a middle-man. This is rarely necessary, but if you are that paranoid, it is available. You can take a :file-transfer automatically from Gozer by typing

dcc-accept-list-add gozer/

Now you will automatically accept files from Gozer after a 15-second delay in which you can choose to deny the transfer. If you take no action, the transfer will start and you will begin downloading the file from their computer. There is no limit to file size whatsoever when using DCC. Send 1kB, or 190 GB, size does not matter, so long as you remain connected to Gozer (or whoever your friend is, we're using "Gozer" as an example). Now you will no longer have to choose "accept" every time they send you a file, and there is never any limit to how many transfers you can take at once, however, you will be taking files into the default folder. To change this to your desktop (for ease of explanation), have Chatzilla open and use the toolbar.

Go to Chatzilla->Preferences....Go to "global settings" and select the DCC tab..Under "Downloads Folder", type-in "file:///C:/WINDOWS/Desktop/" (without quotes) or browse the path to your desktop and all DCC transfers will go there from now on as long as you "apply" the .change before closing the menu

For reliability, it is best to take one at a time though. This is active as long as they keep the same username, usually, even if their IP changes, such as if they use Dial-up/telephone internet access The above "IP" and Dial-Up..." are intended links, so clicking on them will give an explanation) (if you do not know what they mean

:You can PM other users by typing
msg Gozer I think he's lying/

This will send a message only to Gozer: "I think he's lying" (without quotes). No other person will ever see this message, and it will show up in a new tab on the chatzilla client only for the recipient of the message, often with a system-sound to alert the recipient. This is a way to talk privately during a chat with several people and only talk to one person, like whispering to them, but noone can ever know that you did or what you said except who you said it to. Experiment .with it to see how it works

The context is /msg , just be sure to add a space between the username you want to talk to and .the message you intend to type

...Step 4: Operator status and it's priveleges ##

When you are the first user to show up to a channel, you are automatically given "operator status", meaning you are the operator of that channel, or the administrator of that channel. This gives you special privileges that other users cannot use without that status. It is considered polite to give operator status to people you expect to be there with you, so that they can have administrative control of the channel as well. To share operator status with your friend Gozer, :type the following (assuming you were there first)

op gozer/
This command can only be given by someone with operator status to share the same status with a fellow user. What good is "operator status" to the two of you? I'll show you more commands
:that will demonstrate

kick Mr_Rude/
This will kick the intrusive person named "Mr_Rude" from the channel, removing them from the channel. A message can be appended to show a reason why you kicked them, but is not
:necessary. An example is shown below

kick Mr_Rude No swearing is allowed on this channel/
When Mr_Rude is kicked they will receive the message, "You have been booted from
"#instructables by B_Kenobi (No swearing is allowed on this channel)
If you have a persistent user who annoys you and does not respond to the /kick command, you can use the /kick-ban command, that will kick the user off and ban them from rejoining. You can
:also attach a message in the same way as a simple /kick
kick-ban Mr_Rude No swearing is allowed on this channel/

They will see the same message as above, but "booted" will be replaced with "banned", and they will not be able to join your channel as long as you are on it. you can remove this by typing the
:following
unban Mr_Rude/

Now the ban is lifted and that user can log into your channel again. If another operator banned a
.user, you can unban them and override their ban as a fellow operator
Operators can also set the topic for the conversation. To do so, simply type /topic and enter what you want as your topic, to be shown for all users at the top of the screen. Anyone who joins will see this in the introduction to the channel, and all users already logged-on will see the change as
:it is made

!topic I love bunnies/
The topic will change to this after this command is given, if you have operator status. Try it and
...see by looking at the top of the window

If you give someone operator status and they misbehave, an operator can strip another operator
:of his status by using the following command
deop Gozer/

This will remove Gozer's operator status (maybe because he bans your other friends for no reason or abuses his operator status), and remove his ability to change topics, or kick/ban other users. If you wish to not have operator status, you can deop yourself as well, as an operator has control over privileges of all users, including themselves. An amusing way to leave as an operator is to kick yourself. Kick yourself for foul language even when you haven't used any, or
.for any reason. It's an inside-joke amongst veteran IRC users
....It takes practice, but not long before you are perfect with these commands

Step 5: More commands and interesting fun ##
Picture of More commands and interesting fun More fun you can have with IRC is using colors and bold when you type, to emphasise what you are saying, and how you say it. Give your text
:personality by the commands below

By prefixing /me to a statement, you display your text in italics like a footnote or a silent action
:by context. an example is below

Gozer: I really like the new Heidi Klum pix
 :(you type the following in response)
 me hears that/
 :The result is your post showing as
 B_Kenobi: hears that (in italics)
 as if you said it under your breath, or otherwise meant it as a silent thought. Experimentation
 .with this will show how this can be useful in context
 Some of you may wish to type in color or otherwise customise your text. To do so, you need to
 prefix the text with the appropriate symbol. Start the desired text with the percentage symbol
 "%" and a small tool-tip will appear showing the supported commands. For example, %U means
 :to underline the following text. you can end the effect by repeating the command. For example
 !I %Uwant money %U man
 The result is "I want money man", but only "want money" is underlined in that text. Bold is a lot
 :simpler
 I *want money* man
 As typed, "*want money*" will show in bold text. Anything bracketed in asterisks will be shown
 as bold type, but the asterisks will show. Using asterisks are shorthand, but for more perfect
 text, use the %B tag instead so that they won't show. Remember that to use this option, you must
 .use capital letters for every prefix and suffix (not %b, but %B for bold)
 :You can combine tags such as the following
 I %U %B want money %U %B man
 Which will show "want money" in bold and underline at the same time. Skillful tagging can
 show the same in bold/underline//italic all at once. Below I show basic text tagging and it's
 :effect. The commands are as follows (try them when on IRC to see the effect)
 I %Bwant money%B man
 I *want money* man
 I %Uwant money%U man
 I %B %Uwant money%B %U man
 The image shows the difference between these commands. Many commands can be combined
 for the desired effect. Note that you do not need to space between tags and the text as a "text
 tag" (called a "switch") for the client to interpret everything following as the prescribed format
 until terminated, as seen above. A switch (%U or %B is called a "switch", just like how you turn
 the lights on in a room) is set in text, and then terminated to return to the default for following
 text. "%" is a switch, just as "/" is a "command flag". The client responds to "/" as a command
 and "%" as a switch, and this is how you can better control the client. A switch is a trigger for a
 different interpretation of the following text, just as "/" tells the client that the following text is
 .expected to be a command
 I %C9 %U %BWant Money%C9 %U %B
 This will show the latest example in bold, underline, and in green. Spaces between the text and
 the switches are displayed, so if you spaced the tags here, the space would carry the same
 underline. I keep switches in order, but as long as you close them all, the effect is turned-off.
 The order is not specific, just as long as each switch is closed as it was opened, or it remains
 :until closed. Text entered into IRC below produce the same result
 I %C9 %U %BWant Money%C9 %U %B I %C9 %U %BWant Money%U %B %C9

This line has all the same switches, and upon conclusion of the line, all switches were opened, and then closed, returning the text back to its default state. If you omit the %B switch, all your text will be in bold until you "turn it off" on that line..Clients will vary, but chatzilla tends to return to default for the next line. Type the same line above and then add text beyond that. Now try removing a switch, such as "%U"....The effects will continue, but without the underline effect What you see is what you are sending, and another chatzilla client will show the same .thing to who you are sending to

By now, you should be learning something, and are becoming more proficient in computing, even if you are not aware of it. You can get really good with IRC and show-off your skills once .you learn how to control it, and some of this should help you pick-up on these tricks

Now you should be able to use Chatzilla with some proficiency, as well as many other IRC clients. Stop using IE, and start using Firefox, and get the chatzilla extension (no, it's not even on the level of "difficulty", it's child's-play, so you have no excuse), and when you accept the best, you will be above the rest and learn IRC to discover the real IM client that has been aroundlong before YIM or MSN or any other. Use the best, and never need the rest

All you need to do now is to convert your friends, and show them that you know the real thing from the fake. You don't need to install some unstable application just to connect to this .network, you just use the best browser for your internet and your chat-system is right there

If you still think Internet-Explorer is better than Firefox, and refuse to change, forget this Instructables project, and invest heavily in anti-virus programs and everything else you can, and be sure to have a technician on-call to solve your IE-related problems because you choose to be incorrigible. I'll try to feel sorry for you that your 3.1GHz machine is slower than my 1.3GHz machine due to your gaping security-holes that make your machine a zombie for others to enslave, but you choose that by choosing IE, so enjoy the finest of Microsoft programming and the privileges of your inability to accept change despite the evidence contrary to your popular .belief

If you are conservative in software, you automatically fail, so take pride in your failure of self-designed obsolescence and your inability to accept something above the status-quo, because IRC is simply not for you. IRC requires a minimal level of intelligence, so if you think you are going to do this from Internet explorer, just forget it, this project is light-years from your capability or potential skill level. Continue to use IM until you "are all growed-up" and can .handle IRC

The rest of you are already chatting on IRC now, so enjoy, and welcome to the high-tech world in high-tech format. Never use a dumbed-down client again, because now you have real control, and noone has to hold your hand or distract you from what's going on behind your back.

Welcome to the most secure IM system that was always available to you but noone ever told you how. Now you can chat securely without spam or intrusion, and you have easy and .significant control over your chatting area

...Welcome to the 21st century of IM chatting...Enjoy
تیم و مدیریت پروژه
- ساختن یک جامعه ی متن باز - چگونه از IRC استفاده کنیم (Internet Relay Chat) - داستان
یک شخص و ایجاد یک جنبش نکاتی برای مرییان: داستانی از خودت - پنج ابزار برتر مدیریت
پروژه های متن باز - پنج راه برای بالابردن تعامل در جامعه ی متن باز # داستانی از خودت

منبع : [NOI's Story of Self Materials](#)
داستان یک شخص و ایجاد یک جنبش

How can you lead others if they don't know who you are, where you come from, and what your values are? Mastering the art of public narrative allows you to establish firm ground with your constituency in which to collaborate, lead and find common purpose. Each of us has a story to tell that can move others to action. Story of Self is the first part to public narrative where you .will learn to convey why you are called to leadership in your movement

This is from [New Organizing Institute](#)

"?Why am I called to lead"

ACKNOWLEDGEMENTS ##

We welcome your suggestions for improving this guide further for future trainings. We also welcome you to use it and adapt it for your own trainings, subject to the restrictions below. This workshop guide has been developed over the course of many trainings by Liz Pallatto, Joy Cushman, Jake Waxman, Devon Anderson, Rachel Anderson, Adam Yalowitz, Kate Hilton, Lenore Palladino, New Organizing Institute staff, MoveOn Organizers, Center for Community Change staff, Jose Luis Morantes, Carlos Saavedra, Sean Thomas-Breitfeld, Shuya Ohno, Petra .Falcon, Michele Rudy, Hope Wood, Kristen Dore and many others

RESTRICTIONS OF USE ##

The following work [this workshop guide] is provided to you pursuant to the following terms :and conditions. Your acceptance of the work constitutes your acceptance of these terms You may reproduce and distribute the work to others for free, but you may not sell the work to - others. - You may not remove the legends from the work that provide attribution as to source (i.e., "originally adapted from the works of Marshall Ganz of Harvard University"). - You may modify the work, provided that the attribution legends remain on the work, and provided further that you send any significant modifications or updates to marshall_ganz@harvard.edu or Marshall Ganz, Hauser Center, Harvard Kennedy School, 79 JFK Street, Cambridge, MA 02138 - You hereby grant an irrevocable, royalty-free license to Marshall Ganz and New Organizing Institute, and their successors, heirs, licensees and assigns, to reproduce, distribute and modify the work as modified by you. - You shall include a copy of these restrictions with all copies of the work that you distribute and you shall inform everyone to whom you distribute the work that .they are subject to the restrictions and obligations set forth herein If you have any questions about these terms, please contact marshall_ganz@harvard.edu or Marshall Ganz, Hauser Center, Harvard Kennedy School, 79 JFK Street, Cambridge, MA .02138

public narrative: story of self ##

...OBJECTIVES: By the end of this training, you will

Learn the basics of how public narrative works: values, emotion & story structure - Learn - criteria for an effective story of self and coach others on improving their storytelling - Practice and get feedback on your own story of self

EACH OF US HAS A COMPELLING STORY TO TELL ###

Each of us has a story that can move others to action. As you learn this skill, you will be learning to tell a story about yourself, the community you organize with, and your strategy that motivates others to join you in creating change. In addition, you will gain practice in listening, .and coaching others to tell a good story

PUBLIC NARRATIVE IS A PRACTICE OF LEADERSHIP ###

Leadership is about accepting responsibility for enabling others to achieve shared purpose in the

face of uncertainty. Narrative is how we learn to make choices and construct our identities and purpose—as individuals, as communities and organizations, and as nations. What does public narrative have to do with this definition of leadership? You can't ask others to follow you if they don't understand what your intentions are, and why you are called to lead

THE HEAD & THE HEART

There are two ways we understand the world: through our head (strategy & analysis) and through our heart (story & motivation). To enable others to achieve shared purpose, public leaders must employ BOTH the head and the heart of their constituency in order to mobilize others to act on behalf of shared values. In other words, they engage people in interpreting *why* they should change their world (their motivation) and *how* they can act to change it (their strategy). Public narrative is the “why”—the art of translating values into action through stories

VALUES INSPIRE ACTION THROUGH EMOTION

We don't think our values; we feel our values. Often we don't realize what we value in the world until we hear a story or witness an injustice that stirs emotions within us. Emotions inform us of what we value in ourselves, in others, and in the world, and they enable us to express the motivational content of our values to others. Because stories allow us to express our values not as abstract principles, but as lived experience, they have the power to move others to action

:Exploring our values

Each photo below reveals a human story connected to an injustice. Review the photos below to further explore your own values and the process in which you come to realize what you value.

Pay attention to your feelings--what specific emotions do you feel? Did you realize you value something that you may not have articulated before? Do you feel inspired to take action

What feelings or emotions does this photo evoke? | What value does this reveal that you | | hold? | | ---- | ---- | ---- | | HURRICANE KATRINA | | | ICE RAIDS | | | OIL SPILL | | | | | THE ECONOMY

SOME EMOTIONS INHIBIT ACTION, OTHERS MOTIVATE ACTION

Public leaders often encounter individuals or groups where mindful action is inhibited by inertia, fear, self-doubt, isolation, and apathy. The job of a leader is not to tell people to stop feeling this way but rather use storytelling to move people from feelings of stagnation to feelings of motivation - urgency, hope, YCMAD (you can make a difference), solidarity, and anger. The language of emotion is the language of movement—they actually share the same root word. Stories mobilize emotions of action to overcome emotions that inhibit us from mindful action

PUBLIC NARRATIVE COMBINES A STORY OF SELF, US AND NOW

STORY OF SELF

By telling a “story of self” you can communicate the values that move you to lead. Public leaders face the challenge of enabling others to “get” the values that move them to lead. Effective communication of motivating values can establish grounds for trust, empathy, and understanding. In its absence, people will infer our motivations, often in ways that can be very counterproductive. Telling our story of self can help establish firm ground for leadership, collaboration and discovering common purpose

Every one of us has a compelling story of self to tell. We all have people in our lives (parents, grandparents, teachers, friends, colleagues) or characters we love, whose stories influence our

own values. And we all have made choices in response to our own challenges that shape our
.life's path— confrontations with pain, moments of hope, calls to action
The key focus is on our choices, those moments in our lives when our values moved us to act in
the face of challenge. When did you first care about being heard? When did you feel you had to
act? Why did you feel you could act? What were the circumstances, the place, the colors,
sounds? What did it look like? The power in your story of self is to reveal something of those
moments that were deeply meaningful to you in shaping your life—not your deepest private
secrets, but the events that shaped your public life. Learning to tell a good story of self demands
.the_courage of introspection_, and of sharing some of what you find

STORY OF US

By telling a “story of us” you can communicate values that can inspire others to act together by
.identifying with each other, not only with you
Just as with a story of self, key choice points in the life of a community—its founding, crises it
has faced, or other events that everyone remembers—are moments that express the values
shared. Consider stories that members of your group have shared, especially those that held
similar meaning for all of you. The key is to focus on telling a specific story about specific
people at a specific time that can remind everyone – or call to everyone's attention – values that
you share. Telling a good story of us requires the_courage of empathy_ – to consider the
.experience of others deeply enough to take a chance at articulating that experience

STORY OF NOW

By telling a “story of now” you can communicate an urgent challenge we are called upon to
face, the hope that we can face it and the hopeful outcome we can create together, and the
.choice we must make to act now

A story of now requires telling stories that bring the urgency of the challenge alive: urgency
because of a need for change that cannot be denied, urgency because of a moment of
opportunity that may not return. A story of now also offers hope—not make believe hope, but
real, plausible hope, often grounded in what others are already achieving, grounded in the
courage of others' actions, and in the strategic vision of what we can achieve together. At the
intersection of the urgency and the promise of hope is a choice that must be made – to act, or
not to act, to act in this way, or in that. Telling a good story of now requires the_courage of
imagination_, or as Walter Brueggemann named it, a prophetic imagination, in which you call
.attention both to the pain of the world and also to the possibility for a better future

STORY STRUCTURE: CHALLENGE, CHOICE, OUTCOME

Every human story has a plot. A plot begins with a challenge that confronts a character with an
urgent need to pay attention, to make a choice for which s/he is unprepared. The choice yields
.an outcome, and the outcome teaches a moral

A good story allows the listener to empathetically identify with the character and “feel” the
.moral. We hear “about” someone's courage; we are also inspired by it

The story of the character and his or her choices encourages listeners to think about their own
values and challenges, and inspires them with new ways of thinking about how to make choices
.in their own lives

Incorporating Challenge, Choice, and Outcome in Your Own Story

There are some key questions you need to answer as you consider the choices you have made in
your life and the path you have taken that brought you to this point in time as a leader. Once you

.identify the specific *relevant* choice, dig deeper by answering the following questions
Challenge: Why did you feel it was a challenge? What was so challenging about it? Why was it
 ?your challenge

Choice: Why did you make the choice you did? Where did you get the courage (or not)? Where
 did you get the hope (or not)? Did your parents or grandparents' life stories teach you in any
 ?way how to act in that moment? How did it feel

Outcome: How did the outcome feel? Why did it feel that way? What did it teach you? What do
 ?you want to teach us? How do you want us to feel

A word about challenge. Sometimes people see the word challenge and think it means
 describing the worst misfortunes of our lives. Sometimes those are the moments that most
 shaped us. But keep in mind that a struggle might also be one of your own choosing – a high
 mountain you decided to climb as much as a valley you managed to climb out of. Many things
 .may have been a challenge to you and can be the source of a good story to inspire others

VIDEO REVIEW

SAMPLE STORY OF SELF, US, AND NOW

We'll be watching a sample story of Self, Us and Now. While you watch it, think about the
 .elements of SELF – US – NOW that you hear in this story

| SELF | US | NOW |
|--|---|--|
| What experiences and .1
values call this person to
leadership? 2. What choice
points does the speaker include
to show, rather than tell us his
?or her values | Who is the “us” that the .1
speaker identifies? 2. What are
the common values the speaker
appeals to? How? 3. What
challenges and hopes does this
?“us” or community share | What urgent challenge does .1
this speaker identify? 2. How
does he or she make that
challenge real? 3. What gives
us real hope that we can do
something? 4. What is the first
step that each person can take
?to be part of the solution |

**What was the speaker’s purpose in telling these stories? What was s/he moving. 1
 ?people to do**

?What values did this story convey? How? By telling or showing. 2

?What details or images in particular reflected those values. 3

**What were the challenges, choices and outcomes in each part of his story? What. 4
 ?morals do the outcomes teach**

SAMPLE STORY OF SELF: Lilian Molina

.As told at Powershift 2011: a gathering of 10,000 climate activists from around the US<
*Greetings, My name is Lilian Maria Molina and I am the Environmental Justice Director at
 Energy Action Coalition. I am part Mayas-Chorti, Lenca and Palestinian, was born in
 Honduras, Central America and moved to the United States at the age of 5 with my mother. For
 the first couple years my mom and I would take an hour-long ride on a two-floor train; I would
 always rush to the top floor, look out the window, and envision what I would do at our
 destination. I would imagine the cartoons I would watch, salivate over the Kudos and Pringles I
 would be able to eat, and think about all the great toys I would play with. Then one day, as I was
 playing with a fully equipped Barbie Mansion, my mom reached over and handed me a bottle of*

Windex and paper towels; at that moment I realized that our hour-long train ride wasn't a field trip, it was a commute to work. My mom and I were there to clean houses not to play

From that moment on I started to notice that things looked very different in different parts of town. I wondered why some families lived in three floor homes, while I lived in a one-bedroom basement apartment with two families. I wondered why the park equipment in my neighborhood was always broken, but was fancy and new on the other side of town. I wondered if people in the neighborhood where my mom and I cleaned houses had to worry about La Migra coming to their jobs or their homes. I wondered if the kids at these houses ever had to miss school to translate for their parents. I wondered why the police didn't arrest kids around these houses for standing on the corner but my friends back in the neighborhood were arrested all the time. I slowly started to understand that these were two separate worlds

As I got older, I would refuse to take the hour-long train ride with my mom, instead I would hang out with my friends in the neighborhood. When I was 12, my mom noticed that I was starting to get involved in some risky activities. She decided to send me to Honduras for the summer to spend time with Mi Abuelita (grandma). That summer Mi Abuelita, a Natural Healer and Master Gardener, helped me connect to my ancestral roots and taught me how to love nature through gardening. I learned about all the different plants that she used to help heal people and deliver babies - it was an eye-opening experience. That summer I also realized that some of the people that looked like my family and I wore suits to work and lived in houses rather than apartments

When I came back to the U.S, I returned to hanging out with my friends; but when I was 16, I decided I was done watching my friends get beat up, get beat by the cops, or arrested. My friends and I started hosting different activities to keep our friends from joining street gangs.

Throughout high school we organized different events, from parties, to walkouts to bring awareness to the violence in our communities. Around this time I remembered how the garden that Mi Abuelita introduced me to helped me to heal, and started wondering if a garden in our community could have the same impact for other young people. I got super excited and started looking for plots of land around the school. But in my search I learned that most of the land in Little Village was contaminated with industrial pollution. I thought to myself, "You have to be kidding me, on top of all of the issues I was aware of, our land is also polluted? We have poor education, gang violence, police brutality, immigration raids, militarization of schools and we also have contamination in our community? What the heck else could be wrong?" I learned that what my community was experiencing is called Environmental Racism and what we need is Environmental Justice before we can plant gardens here in Little Village...and that is what brought me to the work that I am doing now

Now I am here at Power Shift with Front-line Community Members and our Allies, working with the leadership of front-line communities and helping them create a trans-local movement to oppose corporate power is where there is strategic need for youth leadership

TEAM BREAKOUT SESSION

GROUP PRACTICE WORK

GOALS - Practice telling your Story of Self and get constructive feedback - Learn to draw out and coach the stories of others

AGENDA

Total time: 65 min

- .1 Gather in your team. Choose a timekeeper. Do quick introductions (name and hometown). Articulate group agreements for how you'll work together as during this training. Have your facilitator tell their 2-minute story of self as an example** **.min 10**
- .2 Take some time as individuals to silently develop your "Story of Self." Use the worksheet that follows** **.min 10**
- .3 Choose a partner Practice telling your story of self. - 2 minutes each to tell your story**
 - Focus on the values you want to convey—what specific experiences shaped those values in your life? - Be specific and give lots of detail
 - Use the worksheet "Coaching Tips: Story of Self" to help guide your feedback.**
 - 3 minutes each for feedback:
 - What **values** did the storyteller convey? How specifically? - What is the **Challenge, Choice, Outcome** in each story? - Were there sections of the story that had especially **good details or images** (sights, sounds, smells, or emotions of the moment)? How did those details make you feel? - What could the storyteller do to more effectively convey why they are called to leadership in this ?campaign**min 15**
- .4 As a team go around the group and tell your story one**

.1 Gather in your team. Choose a timekeeper. Do quick introductions (name and hometown). Articulate group agreements for how you'll work together as during this training. Have your facilitator tell their 2-minute story of self as an example
 by one. For each person: - 2 minutes to tell their story - 3 minutes to offer feedback from the group *Make sure your timekeeper cuts you off. This encourages focus and makes sure everyone has a chance to tell their story. Remember, the purpose here isn't to tell a perfect story, it's to practice narrative as part of the work of leadership. | 30 min

.min 10

WORKSHEET

DEVELOPING YOUR STORY OF SELF

:Before you decide what part of your story to tell, think about these questions

?What will I be calling on others to do. **1**

?What values move me to take action and might also inspire others to similar action. **2**

What stories can I tell from my own life about specific people or events that would *show*. **3**

?(rather than tell) how I learned or acted on those values

What are the experiences in your life that have shaped the values that call you to leadership in this campaign

FAMILY & CHILDHOOD

LIFE CHOICES

ORGANIZING EXPERIENCES

Parents/Family
Growing Up

School Career First Experience of organizing
Connection to key books or people

Your Community
Role Models
School

Partner/Family
Hobbies/Interests/Talents
Finding Passion
Overcoming Challenge

Role Models

Think about the challenge, choice and outcome in your story. The outcome might be what you learned, in addition to what happened

Try drawing pictures here instead of words. Powerful stories leave your listeners with images in

__ their minds that shape their understanding of you and your calling

| | | |
|------------------|---------------|----------------|
| CHALLENGE | CHOICE | OUTCOME |
|------------------|---------------|----------------|

:COACHING TIPS

STORY OF SELF

Remember to balance both positive and constructive critical feedback. The purpose of coaching is to listen to the way stories are told and think of ways that the storytelling could be improved

DON'T simply offer vague “feel good” comments. (“That was a really great story!”)

:DO coach each other on the following points

THE CHALLENGE: What were the specific challenges the storyteller faced? Did the 1
?storyteller paint a vivid picture of those challenges

_. When you described _____, I got a clear picture of the challenge “

”? I understood the challenge to be _____. Is that what you intended “

THE CHOICE: Was there a clear choice that was made in response to each challenge? 1
How did the choice make you feel? (Hopeful? Angry?)

_. To me, the choice you made was _____, and it made me feel “

”. It would be helpful if you focused on the moment you made a choice “

THE OUTCOME: What was the specific outcome that resulted from each choice? What. 1
?does that outcome teach us

I understood the outcome was _____, and it teaches me _____. But how does it “
”? relate to your work now

THE VALUES: Could you identify what this person’s values are and where they came. 1
?from? How? How did the story make you feel

_. Your story made me feel _____ because “

It’s clear from your story that you value _____; but it could be even clearer if you told “
”. a story about where that value comes from

DETAILS: Were there sections of the story that had especially good details or images. 1
?(e.g. sights, sounds, smells, or emotions of the moment)

_. The image of _____ really helped me identify with what you were feeling “

Try telling more details about _____ so we can imagine what you were “
”. experiencing

:Record Feedback/Comments from Your Team Members Here

[Section for writing]

”Coaching Your Team's “Story of Self

As you hear each other's stories, keeping track of the details of each person’s story will help you to provide feedback and remember details about people on your team later. Use the grid below to track your team's stories in words or images

:ADDITIONAL RESOURCES

:Videos

[Barack Obama, Keynote Address, "The Audacity of Hope", Democratic National Convention, July 27, 2004, Boston, Massachusetts \(first 7 minutes\)](#)
[NOI Video resource center: Story of Self](#)

:Readings

Jerome Bruner, "Two Modes of Thought", Chapter 2 in *Actual Minds, Possible Worlds*. (Cambridge: Harvard University Press, 1986), p.11 – 25
 Martha Nussbaum, "Emotions and Judgments of Value", Chapter 1 in *Upheavals of Thought: The Intelligence of Emotions*, (New York: Cambridge University Press, 2001), (pp. 19-33)
 George Marcus, *The Sentimental Citizen: Emotion in Democratic Politics*, (University Park: Penn State University Press, 2002), Chapter 4, "Becoming Reacquainted with Emotion" (pp.49-78)
 Malcolm Gladwell, "Small Change: why the revolution will not be tweeted", in *The New Yorker*, October 4, 2010.
 (http://www.newyorker.com/reporting/2010/10/04/101004fa_fact_gladwell)
 Ben Brandzel, "What Malcolm Gladwell Missed About Online Organizing and Creating Big Change", in *The Nation*, November 15, 2010.
 (http://www.thenation.com/article/156447/what-malcolm-gladwell-missed-about-online-organizing-and-creating-big-change)
 منبع : [NOI's Story of Self Materials](#)

نکات اصلی آموزش داستان های ارتباطات بخاطر هیجانات اهمیت دارند: هیجان ما را به سمت عمل می بره یک سری احساسات خاص مانع کرد انجام کار میشه در حالی که یک سری دیگر کمک میکنه به انجام کار. ما می توئیم از داستان هایی استفاده کنیم کخ احساسات باعث آسون تر شدن کار شدن. داستان های خوب ساختار دارند=> چالش، انتخاب، نتیجه روایت عمومی یک نوع از داستانه با سه قسمت : خود' ما' الان داستان شخصی ما یک مجموعه تمرین هست که کمک می کنه به ساخت جامعه با معرفی اینکه اونها کی هستند و چرا از اونها برای همکاری کردن در این کار خواسته شده. ما می توئیم از طریق داستان های شخصی خودمون آموزش ببینیم.

----- | ----- | ----- | | ** مشروح نکات ** | | ** اسلاید ** | | ** بخش ** |
TRAINER STORY (2 minutes | | min 30 مقدمه | | : min 80 مجموع زمان جلسه | | : min 30 seconds) | | Tell your 2 min story of Self, as it relates to this particular training group or campaign. Use the Story of Self worksheet in the participant guide to help develop your story. | |
LEADERSHIP (30 seconds) | | Go back to our definition of leadership. How does public narrative fit into this way of practicing leadership? Stories help us learn to make choices in the face of uncertainty. | | **WHY DO WE TELL STORIES? (1 minute)** | | *Why do we tell stories?*

We tell stories to motivate others to take action with us. We tell stories to enable others to understand who they are and why to take action. Story telling is a leadership art that “enables others” to understand their individual, community and immediate calling. A good story helps us make choices in our own lives, by illustrating how a particular character overcame a challenge. ||

TWO WAYS OF KNOWING (1 minute) || There are 2 ways in which we understand the world around us: 1. The heart understands the ‘why’ (story); 2. The head understands the ‘how’ (strategy). Both lead to action, but we need to learn to lead with the head AND the Heart. Organizing without a strategy isn’t driven and doesn’t lead us anywhere, like walking blind. But having the best strategy in the world doesn’t matter if we’re not motivated to implement it and we can’t get anyone to join us. **Audience Engagement:** *By show of hands, who here has been in an organizing situation that was all about head (strategy)? Like “lets get the petitions signed and then the meeting with the congressional members and then lets make calls, etc.” How’d it go? How about an organization that was all heart- storytelling, community building and no strategy? Emphasize the need for the head **and** the heart.* ||

EMOTIONS, VALUES, ACTION (1 minute) || Stories move us to action by engaging our emotions. Emotions tell us what we value in the world. What are the things that bring you joy & hope? What makes you feel angry? When we hear stories that make us feel a certain way those stories remind us of our core values. We experience our values through emotions. Then we are prepared to take action on those values. **Audience Engagement:** When I told my story earlier, how did it make you feel? What specific emotions did you feel? When? Why? What values did you hear in my story? Would it have been the same if I told you “ I value [insert value]” How did you experience those values? ||

ACTION MOTIVATORS & INHIBITORS (2 minutes) || There are emotions that inhibit action and others that motivate us to act. Inertia, (no forward progress) Apathy (without caring), Fear, Isolation (fail to appreciate interest we share with others) & Self-doubt (I can’t do it)– explain the feelings. **Audience Engagement (Questions to Consider):** * Has anybody felt these emotions before? Have they inhibited you from taking action? Have you seen a community with some of these feelings? What are the actions of a community that has these feelings? How do these feelings change? Just by saying, “don’t be afraid” or “you should care about this!” * Our emotions of inaction don’t go away because some one tells us they should, but because we start to feel other emotions. * Have you been in a moment that your community felt with URGENCY? Has anyone felt that in your communities – ANGER? Have you felt HOPE, SOLIDARITY AND YOU CAN MAKE A DIFFERENCE (YCMAD)? When specifically? * What gives you that Hope, Solidarity, sense of YCMAD? ||

STORY STRUCTURE (1 minute) || **Audience Engagement:** What makes a story? Stories have a common structure. A Plot: something happens to the character, and then the character needs to make a choice, then that choice yields an outcome – and that outcome teaches a moral. What happens when there’s no challenge? What happens when the story is all challenge and there’s no outcome, or hope? ||

3 PARTS OF PUBLIC NARRATIVE (1 minute) || In this session you will be learning how to tell your Story of Self, but since organizing isn’t just about your, but about you in relation to others, you will also be learning the two other parts of public narrative: the Story of US and Now. Three parts to public narrative: 1. Story of Self (why I’m called to this work); 2. Story of Us (why we’re called, why we matter in this effort); 3. Story of Now (what we’re called to do right now) ||

SAMPLE STORY- BARACK OBAMA (15 min total to include 3 min set up, 7 min video and 5 min debrief) || Show DNC 2004 Convention Speech Trainer Tip: Because your audience may have

differing political opinions, make sure to explain that we didn't choose this video because Obama is a Democrat or the President, we use this video because he is one of the greatest speakers in our country and he moved many people to action. This a craft he learned and we are here to teach you the same craft. || | Video Debrief | Debrief: Audience Engagement: How do you feel after hearing that? What specific stories or examples painted the challenge and choice for you? What did you want to know more of? Would it have had the same impact if he had said "My name is Barack Obama, I went to Harvard Law school where I learned x and y. Then I was ran for Illinois State Senate, and won. I really support John Kerry because of x,y,z and you should too?" **Where's the story of self?** 1 Challenge, choice, outcome - focus on choice points (father's journey to US, parents decision to marry, naming "Barack"). 2 Notice that there is nothing about John Kerry in his speech, but rather a story that tells us about his own values. Then the audience trusts him and wants to support who he is supporting. 3. What's the transition? ("My story is part of the larger American story") **Where's the story of us?** 1. Founding story, Declaration of Independence (why did he choose this?) 2. Challenge, choice, outcome, hope. 3. What's the transition? "We have more work to do" **Where's the story of now?** 1. Challenge, choice, outcome, hope. 2. Not statistics, but stories, no 10-point plan || | **LEARNING TEAMS EXPLANATION (5 minutes)** || Review the Story of Self worksheets and learning team agenda before the groups break out. Trainer Tip: Remind the participants to focus on 1 or 2 choice points in your life that teach others about what values move them to be here, and like a good movie their story should be constructed so we can SEE it, HEAR it, FEEL it because **BREAK INTO TEAMS 65 min** | All materials are in the || | **باری کردن** || . of the details participant guide | 1. Review agenda, Set Norms, assign a time keeper (10 min) 2. Develop your story (10 min) 3. Partner pair-and-share (15 min) 4. Share stories in small group (30 min) || | **DEBRIEF 20 min** || Invite 2-3 participants to tell their stories. Have group coach them on making their stories better. Also, define what makes for effective coaching. What coaching questions helped your stories improve? 1. Did they tell a real Story of Self with a clear moment(s) of choice 2. Did they avoid the trap of just talking about the campaign? If not how could they? What in their life are you still curious about? 3. Was there good detail? Where **ابزار** could there be more? 4. Coaching—what types of feedback & questions were helpful? | # 5 برتر برای مدیریت پروژه های متن باز (in 2014)

منبع : [OpenSource.com] (<http://opensource.com/business/15/1/top-project-management-tools->) .Also, [reference 2014's posting](#) .(2015)

Last year, I covered five of the best [open source project management tools](#), like ProjectLibre and OpenProject. The article struck a chord with readers and continues to prove valuable. So, this year I revisited the tools mentioned in last year's article, taking into account comments and suggestions from readers, and provided an update on where they are today. First, I share five new open source project management tools for 2015. All in all, this article will give you a good look at 11 of the top open source project management tools out there

The criteria I used is based on the following: - Is the software provided under an open source license? - Does it have an active community? - Does it have up-to-date documentation available? - Is the source code available? - Are there new or recent releases

Five new tools for 2015

Tuleap Open ALM

[Tuleap Open ALM](#) is not just a project management tool, it is an application lifecycle management tool, including support for agile development and project management. Tuleap received the InforWorld.com Bossie Award in 2013 and is used by Fortune 500 companies, small .and medium businesses, and open source projects

Tuleap allows for [agile, traditional, hybrid, or custom](#) processes for project management. It supports planning, sprints, tasks, reports, and more. This tool is very suitable for open source .development companies, as the tool also integrates with Git, SVN, Jenkins, and more

Tuleap is licensed under a GNU Public License and is available [on GitHub](#). The open source version contains all features and allows for unlimited projects and users. In addition to professional support, there is an active community for support and documentation. Tuleap is in .active development, with version 7.8 in December 2014 being the [latest release](#)

OrangeScrum

[OrangeScrum](#) is a modern project management tool for freelancers, agencies, and small and medium businesses. Features include a [scrum](#) task board, resource planning, progress tracking, and more. It is designed for use by IT companies, education and health services, construction and .manufacturing, and others

The tool is web-based on top of a CakePHP framework. It is available as a free and open source under [GPLv3](#) and available [on GitHub](#). There is also a paid SaaS solution in which the parent .company Andolasoft provides professional services and support

The OrangeScrum developer community has an active [forum and roadmap](#). The latest version is .1.5.1 and under [active development](#)

Taiga

[Taiga](#) is an open source project management platform [in beta](#) for startups, agile developers, and designers. For an introduction, see [Nitish Tiwari's](#) coverage in [Taiga, a new open source project management tool with focus on usability](#). Read more in this [interview](#) with CEO Pablo Ruiz .Múzquiz

Taiga is a tool that aims to solve the basic problem of software usability. Designed with" .this sole aim, the developers claim it's beautiful to look at all day long

With a focus on agile development, Taiga has all of the required features such as a backlog, Kanban, tasks, sprints, and issues. Taiga is open source under the GNU Affero GPLv3 and available [on GitHub](#). There is also a paid version. Taiga runs on Nginx, Python 3.4, and PostgreSQL 9.3 or higher. You can find documentation on their site and support via their Google .group

Odoo

[Odoo](#), formerly known as OpenERP, is a full suite of business applications, of which Odoo project management is just one. It is multiplatform and supports Windows, Fedora, CentOS, .RHEL, and others. Odoo is licensed under [AGPLv3](#) and [available for download](#)

Odoo has a [professional community](#) with everything a user or developer needs, including forums, documentation, IRC, tracker, GitHub, and more. It supports a [Kanban](#) view with tasks and issues as well as a Gantt chart and control deadlines. The tool can easily be adapted to meet your own project management methods. It's also highly collaborative, allowing for multiple members to work on proposals or minutes at the same time and sending tasks as emails .automatically. For the full set of features, check out the [tools description page](#)

MyCollab

[MyCollab](#) supports three modules: MyCollab CRM, Document Management, and MyCollab-Project

MyCollab-Project is the tool we're looking at in this case and has many features, like a Gantt chart and milestones, to time tracking, issue management, risk and problem management. MyCollab-Project comes in [three editions](#), of which the Community Edition is the free option. MyCollab is licensed under AGPLv3, built in Java, and bundles open source frameworks and libraries. The source code is [on GitHub](#) and the tool is available for download [on the website](#). Documentation is available there as well

MyCollab-Project's latest release is 4.5.5 with a focus on enhancing project functionalities and mobile distribution. GitHub is used in place of a traditional forum, where they accept .contributions and provide support

Progress report on tools featured in 2014

.Here's an update on the projects I covered last year

[ProjectLibre](#) has moved from version 1.5.8 to 1.5.9. It has not yet released version 2.0, nor is there a SaaS (software as a service) option available. In July 2014, ProjectLibre passed the 1 .million mark for number of downloads

[LibrePlan](#) current version is the one they released in April 2013. For a full update, check the [.State of the Union 2015](#)

[OpenProject](#) is active and continues to improve as evidenced by their [release notes](#). They are currently at version 4.0.4. The migration to Ruby on Rails 2.1 and 3.2 was successfully .completed in March 2014 with their release of version 3.0

[project-open](#) released a 4.1 beta in June 2014. The expected 4.2 has not been released yet, but I .spotted a version 5.0 on their roadmap for 2015

.[Redmine](#) seems to continue with steady improvements

The bonus tool I covered last year was [Agilefant](#). They have since implemented a paid-for version but continue to provide the basic tool as open source