Pre Fab Academy Program

# Day 1. Introduction

* Welcome
* Presentations
* Digital Fabrication introduction. Fab labs. Fab lab Network
* Fab Lab tour for participants
* Complete survey
* What is Pre Academy
  + Is not Mini Fab Academy
  + Addressed to people who wants to do Fab Academy later
  + Fills Missing gaps
* What is Fab Academy
  + Distributed education
  + Personal Research
  + No roads
  + Free your mind (forget what you already know)
  + Be brave, Take risks (not physical risks)
  + The nine principles. MIT Media Lab
  + Stress control. Can you handle it?
* What it is not
  + Is not a traditional education
  + Tutors vs teachers
* How it works
  + 1 week cycle. Why is it important. Time x effort = constant
  + 19 - 20 weeks, 1 week break
  + Wednesdays 9am in Boston. Global Assignment reviews and Lecture (small bits of info)
  + Assignment
  + Documentation
  + Learn + make + share. Every week different topic.
  + Graduation: Diploma vs Learning
* Start learning now!
  + edX
  + Youtube
  + Instructables
* Mailing list. Do’s and don'ts
  + The lists: class, alumni, instruct
  + Create you local lab communication service
  + Workflow: Research yourself - Local help - Remote Guru - Class list
  + Reply vs Reply all
  + How to ask properly. Going beyond “It does not work. It gives me error”
  + Read. The key to success
* Links to previous years archives and important docs (manual, grading sheets etc.)
* People and entities you should know in the Fab Academy. Their tasks
  + Neil
  + Sherry
  + Coordination
  + Evaluation
  + Regional Reviewer
  + Remote guru
* The final project
  + Remember: it is a 1 week project (production time).
  + Your research may take the whole semester.
  + The Zen of Python <https://www.python.org/doc/humor/#the-zen-of-python>
  + Remember: Fab academy will not convert you into a (you name it)
  + Think about you. Your hobbies and interests
  + Propose what you want to do. Forget how to do it
  + Must be sellable
  + Make meaning. Guy Kawasaki video
    - Increase the quality of life
    - Right a wrong
    - To prevent the end of something good
  + The three types of final projects
    - The good. Simple, well crafted
    - The bad. Overcomplicated, does not solve any problem
    - The ugly. Bad documented, horrible aspect

# 

# Day 2

# Preparing Notebook for academy

* Recommended that each student uses a A4 paper notebook with no detachable sheets. Do not use sheets. Reason is that you WILL loose the sheets.
* What to write down
  + Date
  + Machine used
  + Material (thickness, type, color…)
  + Settings (speeds, feed rates...)
  + Results (what worked, what didn’t)
  + Debugging
* Assignment: Document what you do every day.

# Preparing Laptops and Lab Computers

* The importance of Open Source
* Recommended Backups
  + Backup data to USB
  + Backup data to Cloud
* Showing how to install latest Ubuntu Desktop LTS (currently 14.04)
* Showing how to install software using Ubuntu software center
* Showing how to install software using apt-get
* Showing how to compile software
* Showing how to install windows software using wine
* Showing basic command line for ubuntu
* Assignment: Install list of software
  + “Terminal here” addon for Nautilus
  + fab modules compiled version (deprecated)
    - Instructions in kokompe web site <http://kokompe.cba.mit.edu/>
  + fab modules html5 version (current version)
    - Nodejs <https://nodejs.org/en/download/package-manager/>
    - Install instructions <http://fabacademy.org/archives/2015/doc/fabmodules-html5.html>
    - <https://github.com/fabmodules/fabmodules-html5/tree/master>
  + kokopelli retro (editing Neil’s circuit boards)
    - Install dependencies from kokompe web site <http://kokompe.cba.mit.edu/>
    - Unzip the folder
    - Open folder in terminal
    - make fab
    - sudo make install
    - $: kokopelli -r
  + antimony
  + inkscape
  + openscad
  + gimp
  + cura
  + arduino IDE (not the ubuntu software center version)
  + attiny addon for arduino IDE (Instructions in tutorial section of **Hi Lo Tech MIT Media Lab**)
  + processing
  + qcad
  + git
  + eagle
  + kicad
  + wine
  + partworks in wine
  + Text editor of your choice

# 

# 

# Day 3.

# Creating accounts

It is recommended that students create the following online accounts

* Fablabs.io account
* git.fabcloud.io account for fab academy archive
* Google account
* Youtube and/or Vimeo account for storing large videos
* Github account for your personal code projects
* Dropbox account for sharing files
* Sketchfab.com account for embedding and storing STL files

Assignment: Create all these account, write down all the links

# Repositories

* Install git
  + sudo apt-get install git
* Configure git
  + <https://help.github.com/articles/set-up-git/>
* Create a repository in Github for website
  + <https://pages.github.com/>
* HTTPS link vs SSH link
  + Creating SSH keys <https://help.github.com/articles/generating-ssh-keys/>
  + Change from HTTPS to SSH <https://help.github.com/articles/changing-a-remote-s-url/>
* Workflow for Pulling and pushing
  + git pull
  + git add --all
  + git commit -m “message”
  + git push
* Conflicts. Do’s and Don'ts
  + <https://help.github.com/>

Assignment: Create your repository where you will store your documentation

# 

# 

# Documenting Fab Academy and Pre Academy

* Temporary documentation Index page: <http://thebeachlab.github.io>
* It’s all about documentation. Does not matter what you did, only what you documented.
* Importance of documenting while you work
  + If you are in a trouble, it will help your, local or remote tutor, a friend or the class to understand what you did.
  + It helps yourself during your research to keep track of what you did
  + It helps you to acquire a good habit useful in the future, not only for Fab Academy
* Tell your story, not a step by step tutorial. Is what you did, not what to do <http://fabacademy.org/archives/2013/students/sanchez.francisco/weekly-assignments/index.html>
* Pay attention to the grading criteria and final project requirements
  + <https://docs.google.com/document/d/1v3GzwGlI4TFND0PUZPv_1FMEWwciLfFZj8WpypcbdFI/edit?usp=sharing>
  + <https://docs.google.com/document/d/1Y7sYGcorkyrItmHEYzs4uvXvQC-HigonjJSwO8TYCg0/edit?usp=sharing>
* Documenting in HTML
  + Learning HTML <https://www.edx.org/course/html5-part-1-html5-coding-essentials-w3cx-html5-1x>
  + HTML templates
    - Warning: can distract you from the content
    - If used make sure to remove all Lorem Ipsums and so
    - Useful for starting a new HTML web when you already know HTML
    - <http://startbootstrap.com/template-categories/all/>
  + Software for Editing HTML
    - Text Editor (recommended)
    - WYSIWYG (What you see is what you get). Not recommended. They create messy unusable code.
* Documenting in Markdown
  + Rising tendence
  + Easy to write, fast and clean compared to HTML
  + 10 minutes tutorial <http://markdowntutorial.com/>
  + Ubuntu app ReText
  + Online editor and publisher <https://stackedit.io>
  + Online HTML to Markdown converter <http://domchristie.github.io/to-markdown/>
  + Online Markdown to HTML converter <http://dillinger.io/> (keep both .md and .html files always)
  + Pandoc: Command line converter <https://github.com/jgm/pandoc/releases>
    - Usage ***pandoc -s -o file.html file.md***
    - Create a sh script
* Photography
  + Blurred images
  + Size (web size). There is something even worse than a blurred image, and it is a 16 Mpx blurred image.
  + Crop images
  + Scale down images in right click menu in Nautilus: nautilus-image-converter
  + Backgrounds
  + Composition: Rule of thirds
  + Lighting. Beware of flash.
  + Prepare the scene
* Videos
  + Do not upload videos in the archive. Host them in Vimeo or Youtube
  + Consider GIF for short videos (~5sec)
* Mockups
  + Useful for final project presentation or interface application week
  + <http://www.mockupworld.co/all-mockups/>
* Principles of good design
  + Simplicity <http://vanseodesign.com/web-design/simplicity/>
  + Typography <http://fontpair.co/>

Color harmony <http://paletton.com/>

# Day 4

# Time management

* Supply vs Demand-based time
* serial vs parallel development
* Spiral development
* bottom-up vs top-down debugging (solve the problem or the cause) <https://books.google.co.in/books?id=YmKmWVYqNx4C&pg=PA388&lpg=PA388&dq=bottom+up+vs+top+down+debugging&source=bl&ots=w4Jq3hUaBW&sig=eW5zcSpQQPCwcek_dSDC-Ef7hTo&hl=en&sa=X&ved=0CCgQ6AEwAmoVChMIw5LM6q_TyAIVCpmUCh1hDgI6>

# Preparing the lab

* Cleaning up -Not someone’s else task-
* Inventory
* Access control
  + Facility
  + Sensitive items
  + Machines
* Emergency Plan
  + Fire
  + Police
  + Hospital
  + Pharmacy