

## Installing Arduino, OSC, Python on the chromebook

### SOURCES:

<http://lifehacker.com/how-to-install-linux-on-a-chromebook-and-unlock-its-ful-509039343>

<http://www.howtogeek.com/210817/how-to-enable-developer-mode-on-your-chromebook/>

<https://liliputing.com/2014/06/asus-c200-chromebook-enabling-dev-mode-installing-ubuntu.html>

<http://playground.arduino.cc/Linux/Ubuntu>

<http://www.expertreviews.co.uk/software/operating-systems/1403773/how-to-install-linux-on-a-chromebook-using-crouton>

### PART 0)

Please ensure you have sufficient power and constant internet access throughout this tutorial

### PART 1) Put chromebook into developer mode

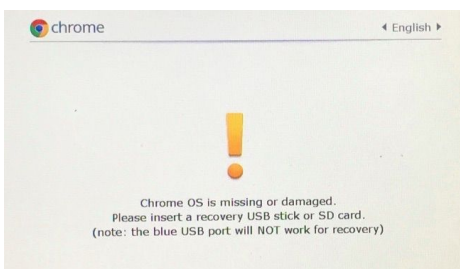
**WARNING:** This will wipe your local data (not that chromebooks can hold much data). It will also violate the warranty

#### 1) Put your chromebook into recovery mode:

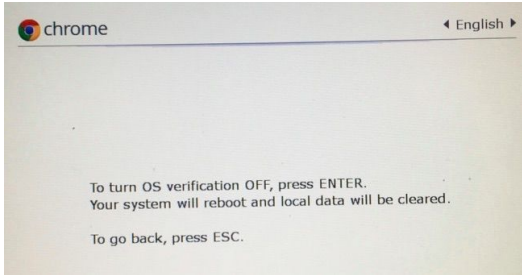
- a) Press and hold the Esc and Refresh keys, then press the Power button (while still holding the other two keys).



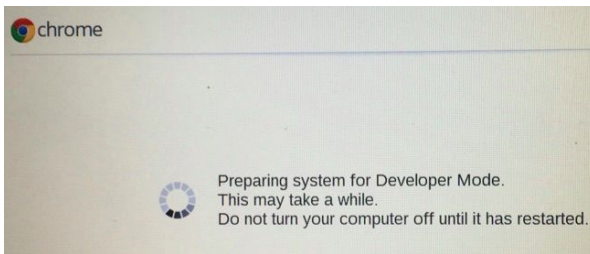
- b) When you see the yellow exclamation point—press Ctrl+D. This will bring up a prompt asking if you want to turn on Developer Mode. (The “Ctrl+D” command is not shown on the screen. Google intentionally hid this feature to prevent people who didn’t know what they’re doing from destroying their chromebooks)



- c) You will see this screen. Press enter to continue.



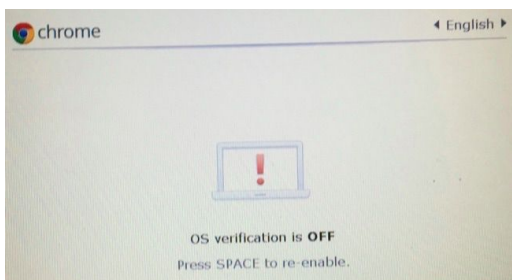
- d) The laptop will start do various things, including restarting several times while beeping obnoxiously. DO NOT TOUCH IT. This will take about 10 minutes.



- e) When you see the welcome page, now you can start touching it. Set up your account.



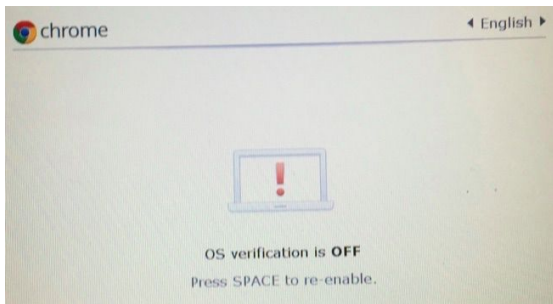
Note: whenever you turn on your chromebook, you will get a message saying that the chromebook is in developer mode. Just wait until the screen goes away, or press "ctrl-d".



**PART 1 - ERROR) The ESC-REFRESH-POWER command isn't working!** Some older Chromebooks have a physical switch that activates Developer Mode.

**PART 1 - REVERSE)** If you ever wish to reset your chromebook (and get out of developer mode)

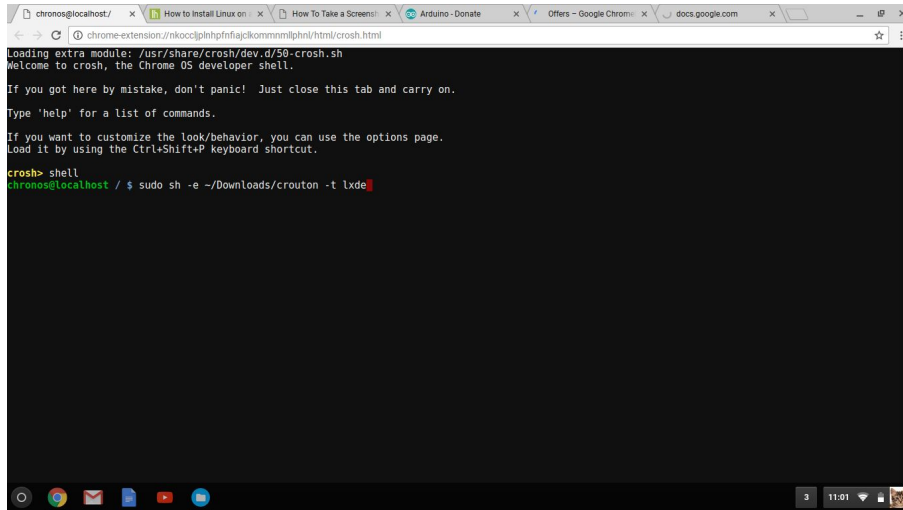
- 1) Restart your computer to bring up this screen



- 2) Hit spacebar followed by enter will get you out of developer mode (and delete everything)

## PART 2) Install Crouton

- 1) Download Crouton >>>[here](#)<<< and save it in your Downloads folder (when you download the file chrome os will save it here by default).
- 2) Press Ctrl+Alt+T to open crosh (chrome's terminal)

A screenshot of a Chrome OS terminal window. The window title bar shows several open tabs: 'chromos@localhost /', 'How to Install Linux on...', 'How To Take a Screen...', 'Arduino - Donate', 'Offers - Google Chrom...', and 'docs.google.com'. The terminal content shows the 'crash' shell interface. It starts with 'Loading extra module: /usr/share/crash/dev.d/50-crash.sh' and 'Welcome to crash, the Chrome OS developer shell.' It then provides instructions: 'If you got here by mistake, don't panic! Just close this tab and carry on.' and 'Type 'help' for a list of commands.' It also mentions customizing look/behavior. The prompt 'crash>' is shown, followed by the user typing 'shell' and pressing enter, which results in a new prompt 'chromos@localhost / \$' and the command 'sudo sh -e ~/Downloads/crouton -t lxde' being entered.

- 3) At the Terminal, run the following command to enter a Ubuntu shell: (for those who don't know how to use terminal, just type in the text below exactly and hit enter). If you get an error at this step ("the shell command does not exist" it means your chromebook is not in developer mode. Go do Part 1.)

**shell**

- 4) Next, run the following command to install Crouton:

MOST CHROMEBOOKS	John Choi's C202s
<code>sudo sh -e ~/Downloads/crouton -t xfce</code>	<code>sudo sh -e ~/Downloads/crouton -t lxde</code>

- 5) Wait about 20 min

```
crosh
Welcome to crosh, type 'help' for a list of commands.
crosh> shell
chronos@localhost / $ sudo sh -e ~/Downloads/crouton -t xfce

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

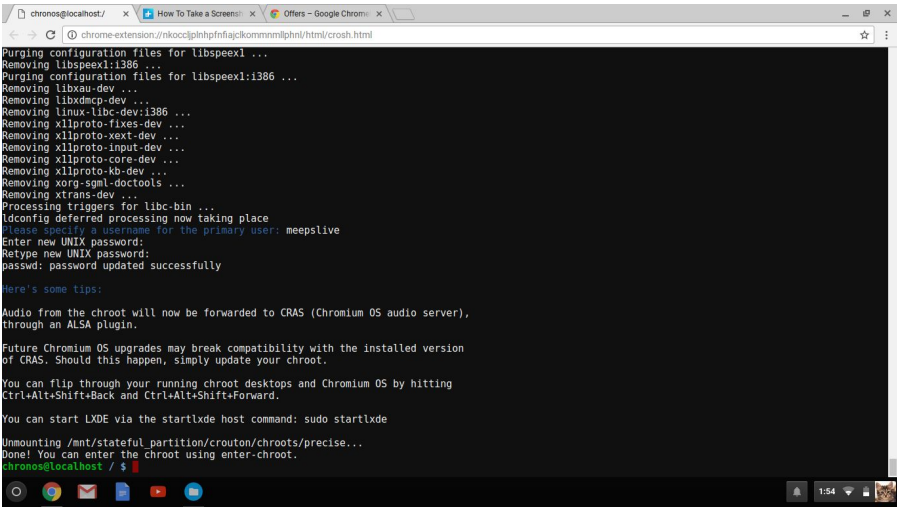
    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

WARNING: Signed boot verification is disabled; enabling it for security.
You can disable it again using: sudo crossystem dev boot signed_only=0
Installing precise-armhf chroot to /usr/local/chroots/precise
Downloading latest debootstrap...
2013-04-29 18:02:35 URL:http://anonscm.debian.org/gitweb/?p=d-i/debootstrap
> "-" [1]
Downloading bootstrap files...
I: Retrieving Release
W: Cannot check Release signature; keyring file not available /usr/sha
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
```

NOTE: this step may fail (several times) due to faulty internet or other issues. Just retry the command until you it asks you for your username

- 6) Follow the prompts to set up your account (type in your desired username once and then type in your desired password twice when it asks for that information)

```
chronos@localhost / $ sudo sh -e ~/Downloads/crouton -t xfce
chrome-extension://nkoocjplnhpfnfajckommnnlphnl/html/crosh.html
libspeexdsp-dev* libspeexdsp-dev:i386* libx11-dev* libxau-dev* libxcb1-dev* libxdmcp-dev* libxext-dev* libxfixes-dev* libx1-dev*
linux-libc-dev:i386* x11proto-core-dev* x11proto-fixes-dev* x11proto-input-dev* x11proto-kb-dev* x11proto-xext-dev* xorg-sgml-doctools*
xtrans-dev*
0 upgraded, 0 newly installed, 25 to remove and 0 not upgraded.
After this operation, 33.9 MB disk space will be freed.
(Reading database ... 42626 files and directories currently installed.)
Removing libc6-dev:i386 ...
Removing libdrm-dev ...
Removing libx1-dev ...
Removing libxfixes-dev ...
Removing libxext-dev ...
Removing libx11-dev ...
Removing libxcb1-dev ...
Removing libpthread-stubs0-dev ...
Removing libpthread-stubs0 ...
Removing libspeexdsp-dev ...
Removing libspeex-dev ...
Removing libspeexdsp-dev:i386 ...
Removing libspeex-dev:i386 ...
Removing libspeex1 ...
Purging configuration files for libspeex1 ...
Removing libspeex1:i386 ...
Purging configuration files for libspeex1:i386 ...
Removing libxau-dev ...
Removing libxdmcp-dev ...
Removing linux-libc-dev:i386 ...
Removing x11proto-fixes-dev ...
Removing x11proto-xext-dev ...
Removing x11proto-input-dev ...
Removing x11proto-core-dev ...
Removing x11proto-kb-dev ...
Removing xorg-sgml-doctools ...
Removing xtrans-dev ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
Please specify a username for the primary user: meepslive
```



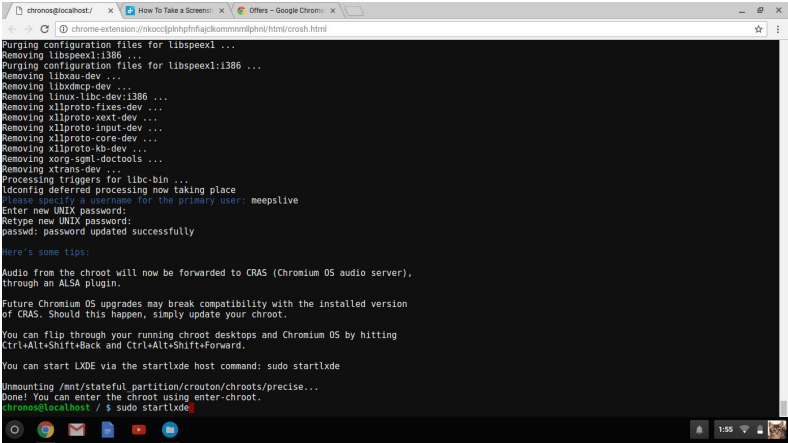
**PART 2 - PRACTICE) familiarize yourself with chronos (optional)**

- 1) Close the terminal (tab) by hitting that X button
- 2) Open it again using Ctrl+Alt+T

**PART 3) Start up Ubuntu**

- 1) After it's finished installing, run the following commands to start your new desktop environment:

Most chromebooks	John Choi's C202s
<pre>shell sudo startxfce4</pre>	<pre>shell sudo startlxde</pre>



**PART 3 - ERROR) OMG I get a scary error message (d-bus) when I try to start ubuntu!**

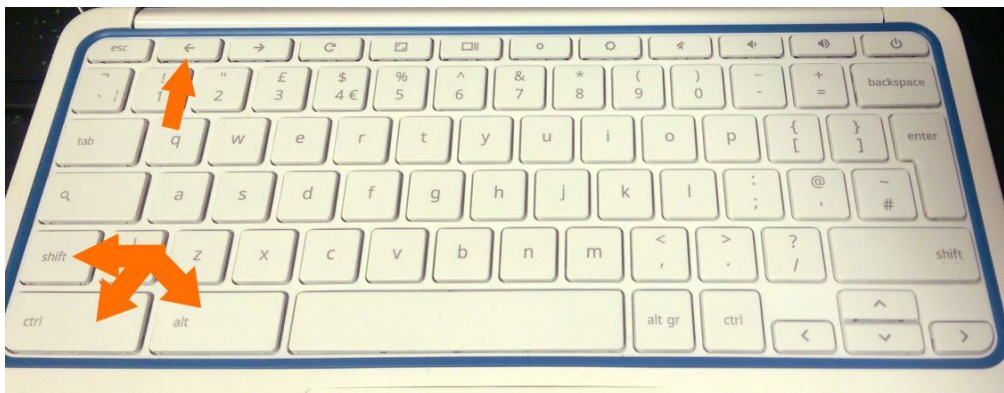
For some reason John Choi's Asus C202s can't handle the default mode of ubuntu that everyone likes. If you get a "d-bus" error, try installing ubuntu in "lxde" mode instead of "xfce" mode.

- 1) Do PART 1- REVERSE) to clean your hard drive
- 2) Redo PART 1)
- 3) Then do steps 2 and 3 again, but follow the instructions on the right

### PART 3.5) Getting familiar with the interface (optional)

- 1) Go back to chrome os and linux using the following keyboard-shortcuts (back and forward keys DO NOT refer to the left/right keys, but rather the left arrow and forward arrow keys you will find at the top of the keyboard)

ARM processors (including John Choi's Asus C202s)	Intel processors
<b>Ctrl+Alt+Shift+Back</b> to go to chrome os <b>Ctrl+Alt+Shift+Forward</b> to go to ubuntu	<b>Ctrl+Alt+Back</b> to go to chrome os <b>Ctrl+Alt+Forward</b> to go to ubuntu <b>Ctrl+Alt+Refresh</b> to bring up the desktop (only for intel processors)



- 2) Exit the Linux desktop by logging out of it like you would on a normal PC—you'll close it completely and go back to Chrome OS
- 3) Use "sudo startxfce4" (or "sudo startlxde") again to log back in.

Note:

-Many Ubuntu tools are left out, so you'll have to install them yourself. Not all apps will be compatible.

-XFCE's screensaver cause graphics issues in Chrome OS and may need to be disabled

-The Downloads folder in Chrome OS is the same as the Downloads folder on the Linux desktop, so if you download or create a file in one environment, you can put it in the Downloads folder to make it available in the other as well.

### PART 4-SHORT) Installing things (short version)

- 1) Open up chronos (Ctrl-Alt-T)

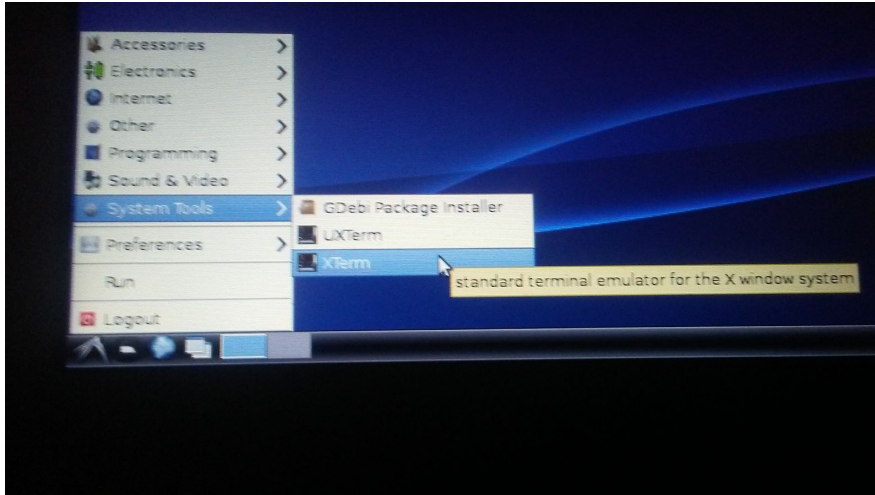


2) Open ubuntu

```
shell
```

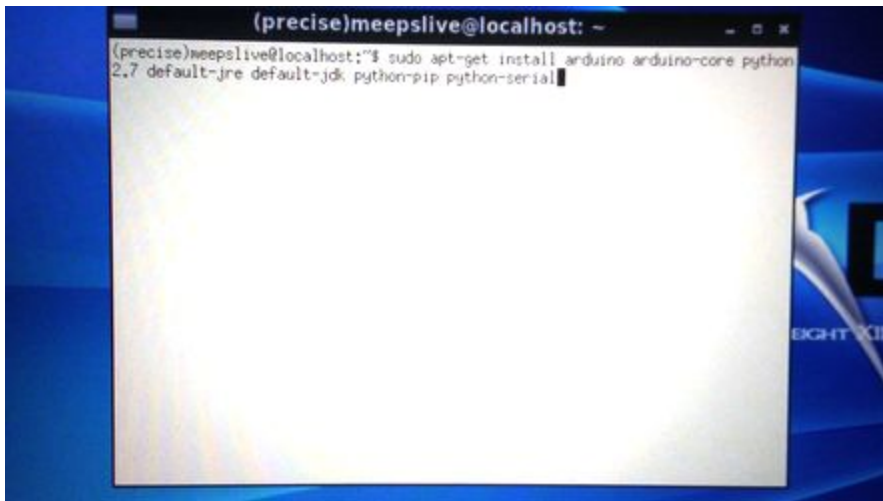
```
sudo startlxde
```

3) Open terminal (may be called terminal, term, xterm, uxterm, etc.)



4) Enter the following to install arduino, python, java, pip, and pyserial

```
sudo apt-get install arduino arduino-core python2.7 python3 default-jre  
default-jdk python-pip python-serial
```



5) Enter your password when prompted



```
(precise)meepsllive@localhost: ~  
(precise)meepsllive@localhost:~$ sudo apt-get install arduino arduino-core python  
2.7 default-jre default-jdk python-pip python-serial  
[sudo] password for meepsllive: █
```

6) Type in “y” when prompted (and wait a long time)

```
arduino-ak avrdude-doc task-c-devel gcc-doc gcc-4.2 equivs libjna-java-doc  
liblcms2-utils pscd pulseaudio libxcb-doc openjdk-6-demo openjdk-6-source  
visualvm icedtea-plugin libnss-mdns sun-java6-fonts fonts-ipafont-gothic  
fonts-ipafont-mincho ttf-wqy-microhei ttf-wqy-zenhei ttf-indic-fonts-core  
ttf-telugu-fonts ttf-oriya-fonts ttf-kannada-fonts ttf-bengali-fonts  
python-distribute python-distribute-doc python-wxgtk2.8 python-wxgtk2.6  
python-wxgtk  
The following NEW packages will be installed:  
arduino arduino-core avr-libc avrdude binutils-avr ca-certificates-java  
default-jdk default-jre default-jre-headless extra-xdg-menus gcc-avr  
icedtea-6-jre-cacao icedtea-6-jre-jawm icedtea-netx icedtea-netx-common  
java-common libasynct0 libatk-wrapper-java libatk-wrapper-java-jni libflac8  
libftdi1 libice-dev libjna-java libjson0 liblcms2-2 libnss3-ld libpcsc-lite1  
libpthread-stubs0 libpthread-stubs0-dev libpulse0 librtx-java libsw-dev  
libstdc++6 libvorbisenc2 libwrap0 libx11-dev libx11-doc libxau-dev  
libxcb1-dev libxdmcp-dev libxt-dev openjdk-6-jdk openjdk-6-jre  
openjdk-6-jre-headless openjdk-6-jre-lib python-pip python-pkg-resources  
python-serial python-setuptools topd ttf-dejavu-extra tzdata-java  
x11proto-core-dev x11proto-input-dev x11proto-kb-dev xorg-sgml-doctools  
xtrans-dev  
0 upgraded, 57 newly installed, 0 to remove and 0 not upgraded.  
Need to get 87.3 MB of archives.  
After this operation, 191 MB of additional disk space will be used.  
Do you want to continue [Y/n]? █
```

7) Enter the following to install python OSC drivers (no need to enter password or type “y”)

```
sudo pip install python-osc
```

Note: you may need to type in “sudo” (when using command line) or select “run as administrator” (when using graphic user interface) when running various programs, including arduino to give the software full permissions.

**PART 4-SHORT-ERROR)** if the above commands do not work, install things individually by following the instructions below

## **PART 4.1) Installing arduino**

- 1) Open ubuntu
- 2) Open up terminal in whatever version of ubuntu you installed (it will be in the start menu) and type in the following:

```
sudo apt-get install arduino arduino-core
```

- 3) Type in password when prompted
- 4) Type in the following to open arduino (the “sudo” gives you write access to the serial ports)

```
sudo arduino
```

- 5) Type in password when prompted
- 6) ENJOY

**PART 4.1 - ERROR) You get an error saying that you need to be added to a “dailout” list and that you need to logging out and logging back in so you can upload code to an arduino. Then, you will be unable to access the serial ports.**

If you get this error, ensure you typed in “**sudo**” at the front of your command.

(Hitting “add” and then logging out and logging back in doesn’t fix the problem, although if the problem persists then there’s no harm in trying it anyways.)

**PART 4.1 - ERROR) Permission denied. You can open up arduino, and see the serial ports, but you can’t upload code.**

This may happen if you try to install arduino normally using the GUI. Installing using the GUI prevents you from using “**sudo**” when you open it, meaning you will not have permissions.

## PART 4.2) Installing python and other developer package

### Note:

-Ubuntu comes installed with both python 2.7, python 3.5 and the java runtime environment (but not the java development package.)

-Everything from here onwards is not chromebook specific, and can be found by searching "how to install <program> on ubuntu"

- 1) Open ubuntu
- 2) (optional) Update your linux

`sudo apt-get update`

- 3) Find the instructions below specific to what you want to do

	Terminal (easy way)	Manual (flexible way, go to the link below and download the latest version of what you are trying to download)
Python 2.7	<code>sudo apt-get install python2.7</code>	<a href="https://www.python.org/downloads/source/">https://www.python.org/downloads/source/</a>
Python 3.5	<code>sudo apt-get install python3</code>	<a href="https://www.python.org/downloads/source/">https://www.python.org/downloads/source/</a>
Java	<code>sudo apt-get install default-jre default-jdk</code>	
Java runtime (for running java on web browsers)	<code>sudo apt-get install openjdk-7-jre</code>	<a href="https://apps.ubuntu.com/cat/applications/openjdk-7-jre/">https://apps.ubuntu.com/cat/applications/openjdk-7-jre/</a>
Scratch		<a href="http://learn.linksprite.com/scratch/how-to-install-scratch-on-ubuntu-14-04/">http://learn.linksprite.com/scratch/how-to-install-scratch-on-ubuntu-14-04/</a>

**STEP 4.2 - ERROR)** Try running your program with sudo.

#### STEP 4.3) Install OSC (Open Sound Control) Python Plugin

- 1) Open ubuntu
- 2) Install pip

```
sudo apt-get install python-pip
```

- 3) Install Open Sound Control

```
sudo pip install python-osc
```

#### STEP 4.4) Install Pyserial

- 1) Open ubuntu

Python 2.7	Python 3.5
<pre>sudo apt-get install python-serial</pre>	<pre>sudo apt-get install python3-serial</pre>

#### **STEP 5) Serial on chrome OS (native, in case you're too lazy to install/open ubuntu)**

- 1) Go to the link below

<https://chrome.google.com/webstore/detail/arduino-chrome-serial-mon/opgcocnebgmkhcafcclmgfldjhlncjd?hl=en>

#### **STEP 6) Python on chrome OS (native, in case you're too lazy to install/open ubuntu)**

- 1) Go to <https://repl.it/languages/python>
- 2) Enjoy

This method is highly restrictive because it does not have access to your drivers and the like. Thus, if you want to use pyserial, you will need to do it on ubuntu.

## **Other options:**

### **Skulpt Interpreter:**

<https://chrome.google.com/webstore/detail/skulpt-interpreter/bocjplmmdjglmffmpofmmndklbdpcmeb?hl=en&gl=US>

Runs python within browser

### **OSC on native chrome os:**

<https://www.npmjs.com/package/osc> < public webpage

<https://github.com/colinbdclark/osc.js/> < github source code

This claim to work on chrome os. Have no idea how to use javascript or OSC so haven't tried it.

### **PlatformIO:**

<http://platformio.org/lib/show/1103/ArdOSC>

Claims to work on any platform, has OSC libraries. They got it working on arduino, and ESP8266, so no reason why they couldn't get it working on chrome os.

### **Photon: (highly recommended) (\$20)**

<https://store.particle.io/#photon>

This device can be communicated through online, including through IFTTT.

<https://console.particle.io/>

The easiest way to upload code to the photon is through the online code editor.

<https://build.particle.io/>

## **Things that don't work:**

### **Python Shell on native chrome os:**

<https://chrome.google.com/webstore/detail/python-shell/gdiimpmmdoofmahingpgabiikimjgca?hl=en>

Basic python interpreter. Can't import modules.

### **Python on native chrome os:**

<https://chrome.google.com/webstore/detail/quickpy-python-interprete/fefdekpmfkgenlofhigjalhkbbedmel?hl=en>

Basic python interpreter. Can't import modules.

### **Codebender: (programming arduino on web browser)**



this was very popular, but closed down recently. You can find a version of it that was working online but that only lets you do 30 lines. You can achieve the same effect by using a particle photon (edit code + upload on browser) but it requires a physical photon.

### **Chromeduino:**

<https://chrome.google.com/webstore/detail/chromeduino/dmkincdpchiadkhocmbpjijebfifgbl>

have not been able to get this working with a chromebook, possibly a chromebook-specific issue. Furthermore, it is not a formal project and is lacking many features

### **Awesome Terminal: (serial terminal)**

<https://chrome.google.com/webstore/detail/awesome-terminal/klldcepbkgbdfacicjceendjjklpof>

May be useful for debugging if anyone can figure out how this works

### **Pyserial on native chrome os:**

Can't find anything. Any search that contains "python" "serial" and "chrome" all point to running pyserial on ubuntu.

### **Other Chrome Apps:**

BlockyDuino: Doesn't let you upload

Arduino to Dataino Serial Proxy: Doesn't start up

WFDuino for Scratch: requires BT

Wylidrin Studio: requires Arduino Yun

AVRChick: doesn't upload. Maybe I'm using it wrong, but it also doesn't have a built-in editor

Diginow Serial Monitor: Frequently fails to start up (only solution is to restart chromebook)

Python by Sam Clegg: Crashes immediately after starting