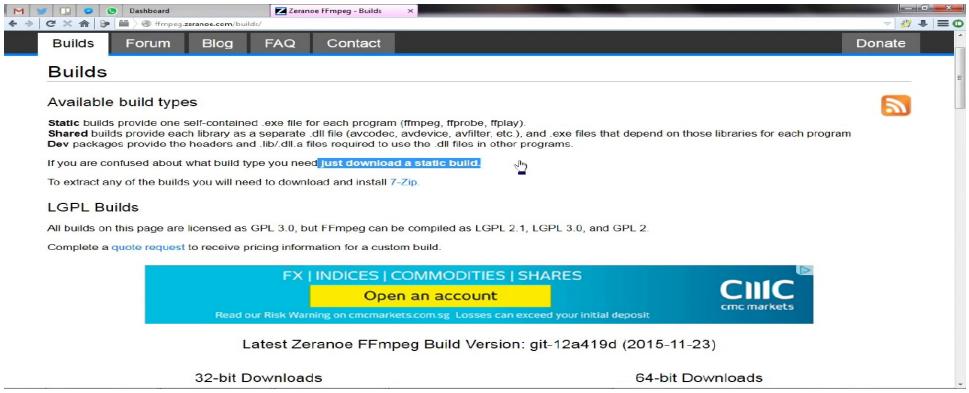
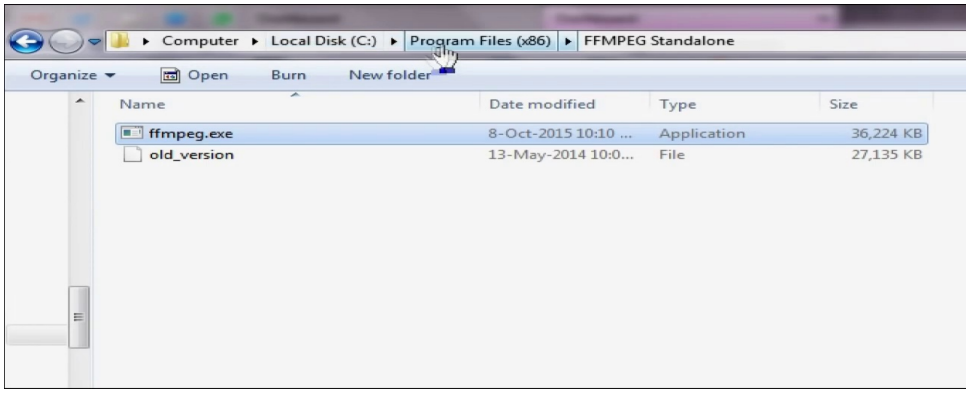
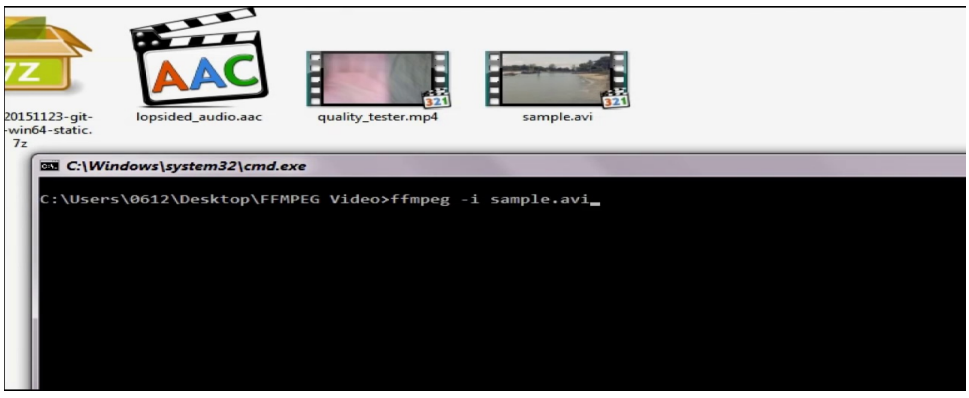


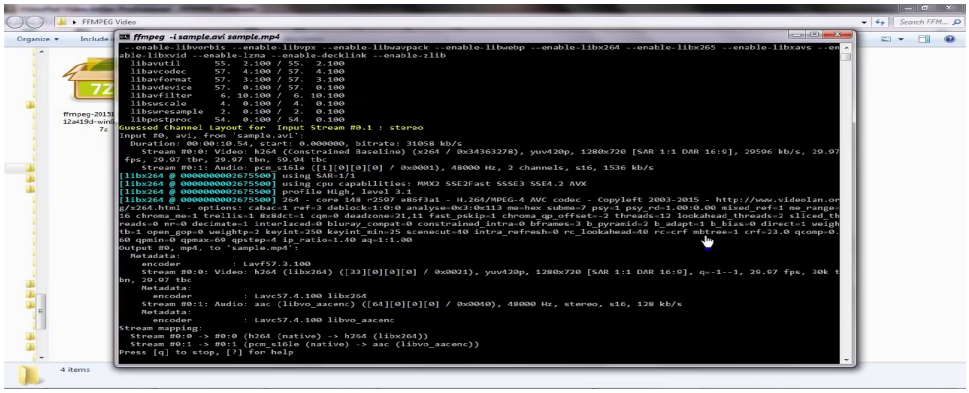
Ffmpeg is actually a command line utility and its job is to do a file format conversion and in that process it can also do a number of cool things now because the license of impact it is actually included in certain other pieces of software you know to do its job in the background, and as a result you may or may not realize that it's actually that, but in fact it is. That is the plain vanilla FF mpeg running on your computer from scratch. I've actually found it quite useful and indispensable in many situations. You're watching another random Wednesday episode on 0612 TV. So Ffmpeg very interesting tool. We'll then move on to the basics and then to some more interesting things you can do with this piece of software.But yeah, like I said, we'll see it when the time comes first, of course, is how do we install this program? It will send you to the download page on your official FF MPEG site. Go ahead and download the latest version for whatever business your computer.Open the archive in an archive manager of some sort if you don't have one, you can check out peazip, which I will also link to in a video description. That's really the only file you need, so extract that to some location that you can remember and we can begin converting now if you intend to use Ffmpeg from anywhere on your computer, you might want to, you know, save it somewhere important, like.



So what you're seeing on screen right now are the steps I've used to do that. You just tell it what follow the process and what kind of output you want and it just does it. So let's actually try a very simple example.Now what you see here is a folder in which I have staged the files I want to work with. Then hold down shift, right click, empty space, and then click on open Command window here. Here's what Ffmpeg syntax looks like. Then you say dash I, which means I'm about to specify an input file and then you go ahead and specify the input file.

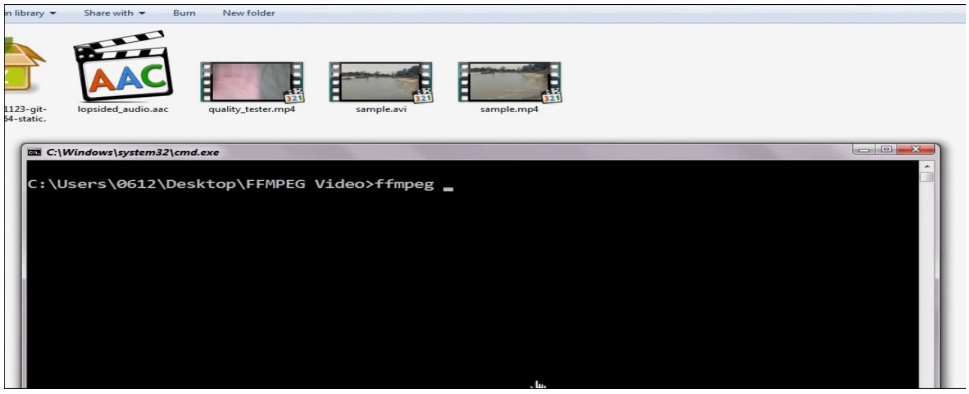


So simply all you have to do is to actually press space and type out the new name of the file, making sure that the extension is MP4, hit enter and well off. FFM Peg is now converting an Avi file to MP4.

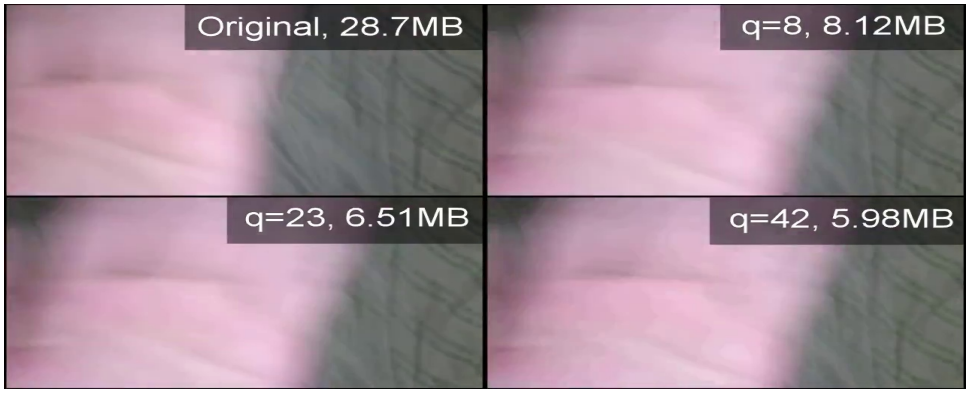


But well, the time will come when you want to do something more complex, and that is where the command switches start to commit. You would probably want to be able to tweak up the quality and what I'm going to show you here works for MP4 and Avi respectively.

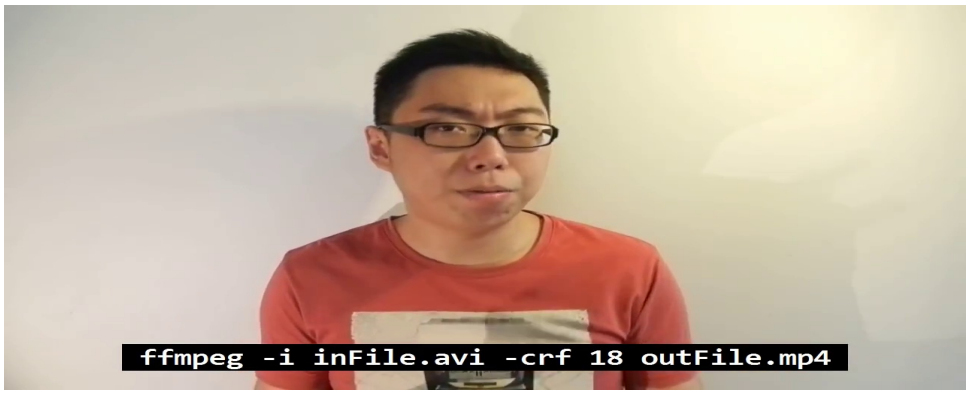
To set the quantizer for an output Avi file, all you have to do is what you call Ffmpeg.



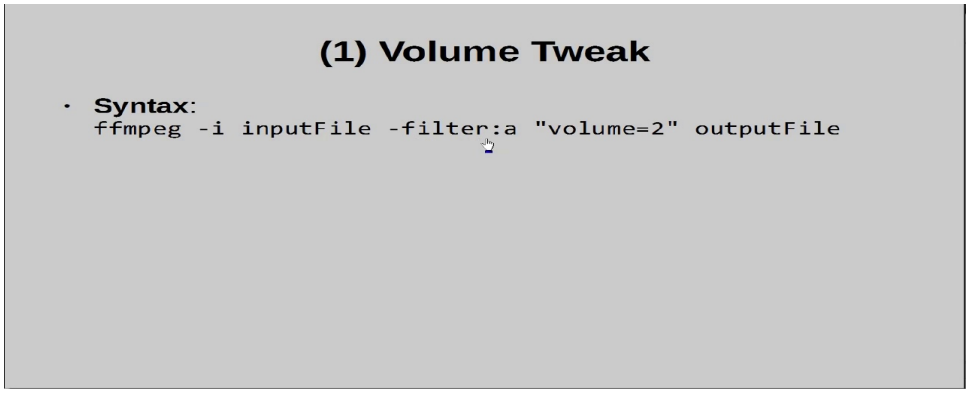
This number is of course the quality setting you want. You'll be able to see the difference in the different settings you choose.



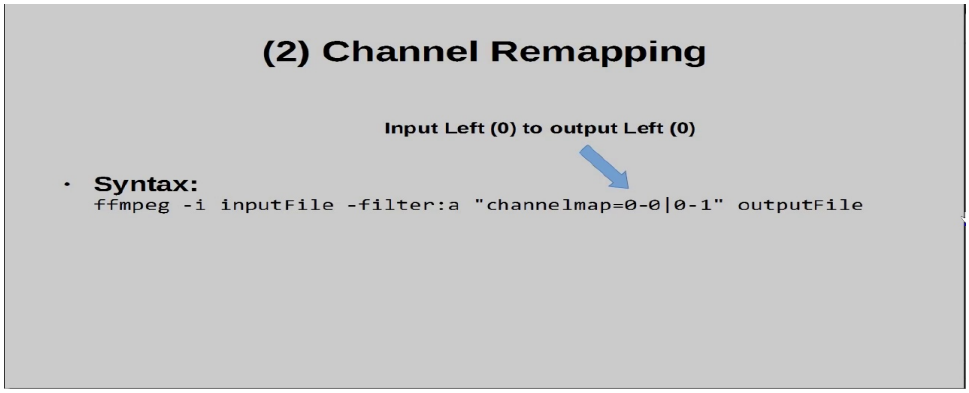
If you're doing this for MP4 files, then you want to specify the CRF value instead, so instead of dash Q, you type dash CRF and then enter your number. The values are not necessarily the same. Look at the file size.



And decide whether you want to bump it up or down. Now bitrates work for both audio and video in a file, so you'll probably want to tell it whether you want this to work on the video channel or the audio channel. What I've done is through the whole list of video and audio filters, I've picked a total of. Two of them are for audio and the remaining three are for video, and these are the things you probably do the most often. We're going slide show mode once again because we've overrun.First, let's tweet the volume.

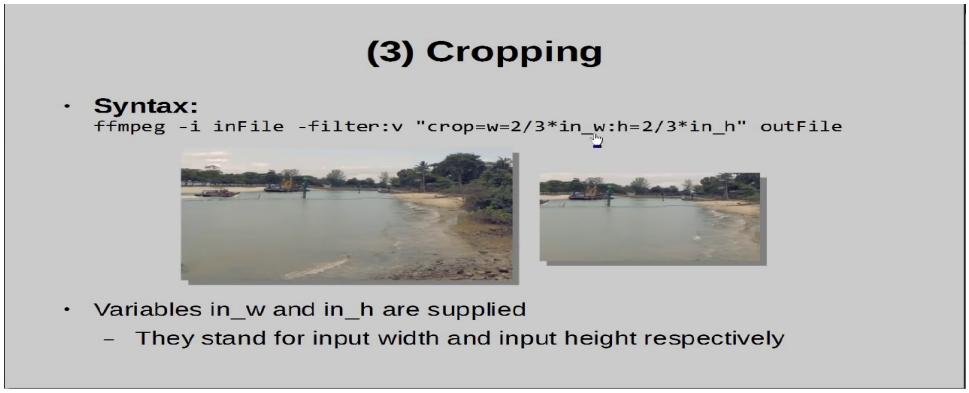


The flag itself says we want to use a filter on the audio channel and its parameter says that we want to call the volume filter with a setting of two. So when we say volume equals two, we are actually doubling the volume.Next channel remapping.We can easily fix this by mapping the audio from the left input channel to both the output channels. So what we're doing here is mapping the input left channel to the output left channel, and then the input left channel to the output right channel, thus creating an audio track that has sound on both sides.

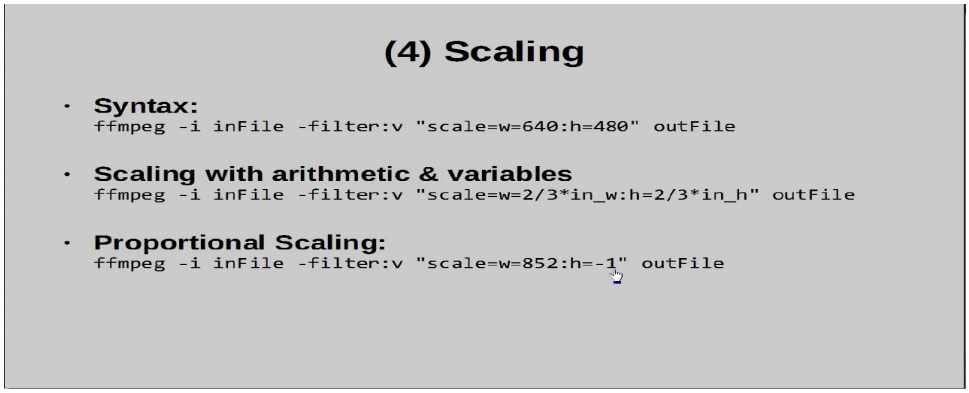


The syntax looks like this.We then say crop and specify up to four parameters.The two remaining parameters are optional, and these specify the upper left corner of the cropping.You can use arithmetic for this as well.

You can use arithmetic for this as well.The variables in_W and in_each are available referring to the input width and height, respectively.



This looks very similar to the first two parameters of the crop filter used to set the width and height of the output.In addition, if you like proportional scaling, you can specify negative one for one of the input parameters.



Finally, rotation simply say rotate equals and specify your angle.Luckily, it's not that hard since you can just multiply it by the constant Ï€ / 180.The result of this operation looks something like this. We've actually gone through a pretty comprehensive beginners guide to installing and using FF, MPEG, and with even taking a look at some complex features in the form of some filters.

Like I said, FF MPEG has a whole host of filters and if you want to look at the full list, I will include the link in the video description so you can find out more. It does look like I've overrun a bit, so sorry about that, but hopefully you found this video insightful.