Use a USB stick as a CD to copy an operating system ISO.

Beware, this will destroy any info you may have saved on the USB

Prepare a cup of coffee, and place it to one side,

Change directory to where the **ISO** image is saved.

```
cd ~/Downloads
```

Determine the disk configuration before you insert the USB

```
ls /dev/sd*
```

/dev/sda /dev/sda2 /dev/sda4 /dev/sdb /dev/sdb2 /dev/sdb6
/dev/sda1 /dev/sda3 /dev/sda5 /dev/sdb1 /dev/sdb5 /dev/sdb7

Insert the USB now look where your USB has been installed.

```
ls /dev/sd*
```

/dev/sda /dev/sda2 /dev/sda4 /dev/sdb /dev/sdb2 /dev/sdb6 /dev/sdc /dev/sdc1
/dev/sda1 /dev/sda3 /dev/sda5 /dev/sdb1 /dev/sdb5 /dev/sdb7 Cut 2 /dev/sdc2

Your USB is /dev/sdc

Do an **ls** to identify the filename of the required **ISO**, then and then copy it

ls

linuxmint-18.3-cinnamon-64bit.iso linuxmint-18.3-kde-64bit.iso
Cut 1

Now construct the dd command using cut and paste

sudo dd if=linuxmint-18.3-cinnamon-64bit.iso of=/dev/sdc

| |
Paste 1 Paste 2

Execute the **dd** command.

Note. the **dd** command does not provide feedback while it is running.

Sit-back, enjoy your coffee and wait.

When it is complete, if you check out your USB you will notice it now has only one partition.

ls /dev/sdc*

/dev/sdc

Note: all info on your USB has been destroyed except for your ISO

There are a lot of different tools such as Unetbootin and Image Writer that exist for creating bootable Linux media on a USB device. What many newer Linux users may not be aware of is that if they are creating their bootable Linux media from an existing Linux system (or another Unix based OS such as OSX) their computer has a built in command for creating a bootable USB device. This is going to be a short tutorial on how to create a bootable USB drive for your favorite Linux distribution.

First things first you are going to need to have a copy of the ISO image you want to make a bootable media of and a flash drive that is as large (or larger) than the ISO image. For this example I am going to use the latest Bodhi 3.0.0 32bit ISO image (which you can find here) and I am going to assume the ISO image is saved in my user's **Downloads** folder.

To start we need to determine the location of the drive we want to write to. **Before** you attach the USB drive to your computer run open a terminal and run the command:

```
ls /dev/sd*
```

You will see an output that looks something like (but not exactly) like this:

/dev/sda /dev/sda2 /dev/sda4 /dev/sdb
/dev/sda1 /dev/sda3 /dev/sda5 /dev/sdb1

These are the drives that are already attached to my computer. I do **not** want to write the Linux image over one of these existing drives.

Next I plug in the USB drive I want to write the image to and run the above command again. This time the output looks like this:

/dev/sda /dev/sda2 /dev/sda4 /dev/sdb /dev/sdc
/dev/sda1 /dev/sda3 /dev/sda5 /dev/sdb1 /dev/sdc1

The **/dev/sdc** drive that has now appeared when I attached the USB drive is the device I want to write my image to.

Next I need to change directories to where my ISO image is saved to. In this case I have it saved in my Downloads folder so I will run the command:

cd ~/Downloads

Next I will run the command that writes the ISO image to the USB device. Please note that running this command will **erase all data from your USB drive**.

```
sudo dd if=bodhi-3.0.0-32.iso of=/dev/sdc
```

A couple of things to note about this command. First is that this command needs to be run as root, so we have prefixed it with the **sudo** command. Second is that the path to the USB drive we want to write to does **not** contain a partition letter. Meaning it is simply **/dev/sdc** not **/dev/sdc1**. Lastly, writing images takes some time and the **dd** command does not provide feedback while it is running. So grab yourself something to drink while it is working and just let it do its thing.