Lav din egen webserver med raspberry pi computeren.

**Setup Raspberry Pi**

Hent Raspbian (Her. <https://www.raspberrypi.org/downloads/raspbian/>) Eller

NOOBS (<https://www.raspberrypi.org/downloads/noobs/>)  
(TRICKS: Opret en fil der hedder SSH på rod drevet, så kan man logge på via Putty (<https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>), så behøver man ikke keyboard og skærm)

Hent Rufus (Windows) <https://rufus.ie/>

Hav et Mikro SD Klar til at installere på .

Når installationen er udført så sæt kortet i Raspberry pi og sæt strøm til.

Lav en statisk IP-adresse.  
Skal du lege med en test computer, er dette skridt ikke nødvendigt. Skal du derimod bruge den til andre opgaver, er det en god ide, at sætte en statisk IP-adresse, eller kan du risikere den skifter adresse, hvis netværket genstarter.

Skriv  sudo nano /etc/dhcpcd.conf

interface eth0  
static ip\_address=10.0.0.100  
static routers=10.0.0.1  
static domain\_name\_servers=75.75.75.75 75.75.76.76 2001:558:feed::1 2001:558:feed::2

interface wlan0  
static ip\_address=10.0.0.99  
static routers=10.0.0.1  
static domain\_name\_servers=75.75.75.75 75.75.76.76 2001:558:feed::1 2001:558:feed::2

Ovenstående kan man under static IP sætte hvilken IP-adresse man vil have på WiFi (wlan0) eller kablet (eth0)

**Setup software**

Nginx (Webserver softwaren)

Log på raspberry pi og skriv følgende.

sudo apt-get update (Her opdatere vi computerens liste af software som er tilgængelig)

sudo apt-get install nginx (Her installere vi Nginx webserver softwaren)

sudo service nginx start (Servicen nginx skal vi sikre os virker, så den skal startes)

sudo service nginx status (Her bekræfter vi, at servicen virker)

systemctl status nginx.service (Her bekræfter vi, at servicen virker)

Mariadb (Database)

Nu skal vi installere databasesoftware mariadb

Skriv I en terminal:

sudo apt-get install mariadb-server mariadb-client (Installere softwaren mariadb)

Nu skal vi lave en database klar til at modtage data fra vores webside.

sudo mariadb (Vi logger på database med administrator rettigheder)

DROP USER 'root'@'localhost' (Her sletter vi root adgangen til administrator brugeren og gør klar til ny bruger)  
CREATE USER 'root'@'localhost' IDENTIFIED BY 'toor'; (Her opretter vi ny bruger, du kan selvbestemme hvad der skal stå under bruger og adgang, bare husk det, for du skal bruge det senere)

GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'localhost'; (Her sætter vi rettighederne, **BEMÆRK er ikke sikkert nok til internetbrug)**

CREATE DATABASE wordpress; (Nu opretter vi databasen “wordpress” som vi skal bruge til wordpreess installationen)

**Installation af PHP**

For at wordpress skal virke, skal vi have PHP installeret, så vores lille server kan vise hjemmesider skrevet i PHP.

Sudo apt-get install php7.0 php7.0-fpm php7.0-mysql (Installere PHP og Mysql til php)

sudo apt-get install mysql (Selve database software bliver installeret her)

**Installation af wordpress**

Skriv følgende for at få installeret Wordpress

cd /var/www (Her navigere vi til www folderen på webserveren. www folderen er de filer der vises på nettet)

sudo wget <https://wordpress.org/latest.tar.gz> (Her downloader vi wordpress filerne )

sudo tar xvfz latest.tar.gz (Vi pakker filerne ud)

sudo rm latest.tar.gz (Vi fjerne de filer vi ikke skal bruge)

sudo mv wordpress hjemmesideblog (Vi flytter filerne til mappen hjemmsideblog)

sudo chown -R www-data:www-data /var/www/hjemmesideblog/ (Vi ændre rettighederne til mappen, så de kan benyttes på hjemmesiden og den virker med rigtige rettigheder. )

Skriv 127.0.0.1 i en browser og følg vejledningen herfra.

Der skal du bruger adgangen til databasen du oprettede tidligere.

MERE KOMMER

Video

<https://www.youtube.com/watch?time_continue=189&v=Hxoa_t61TtY>