Raspberry Pi – write your own network client!

Application programs on computers have to be run in order to take advantage of the network – a ***server*** listens for incoming connections, and a ***client*** makes outgoing connections. When a client connects to a server, the server ***accepts*** the connection and performs some service. The ***World Wide Web*** is an example of using a network application – the client is the ***Web Browser*** (e.g. chrome) and the server is the ***Web Server*** (e.g. apache)

A simple client

Type in the following program using the IDLE 3 editor, and save it as `client.py`

Make sure that you get the indents correct, as they are important to the meaning of the program.

```
import network
import time

network.call("localhost")

while network.isConnected():
  print("sending")
  network.say("hello")
  time.sleep(1)
```

Testing your Program

First we will test this client just on your own Raspberry Pi.

Open a lxterminal window and run your server:

```
python3 server.py
```

Open another terminal window and use your client to talk to your server

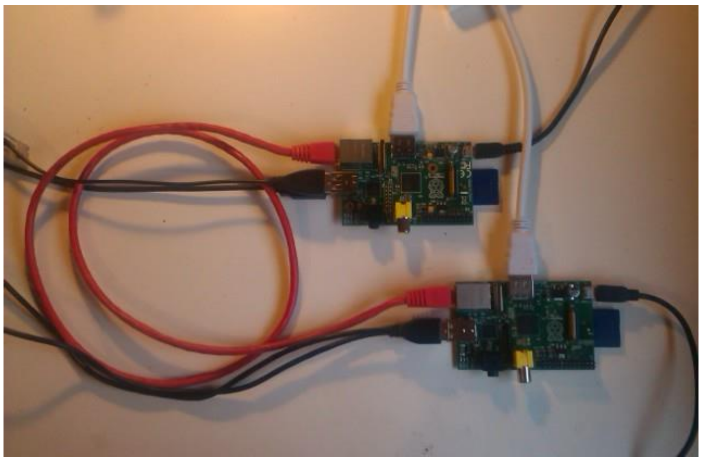
```
python3 client.py
```

You should now see a message come up on both the client and server windows once every second, to show that the message has been transferred between the two programs.

Connecting two computers

The last thing to do is to change the line in client.py program that says "localhost" and put in the IP address of the other computer you want to talk to – then run it again and see what happens!

You will need to know the IP address of the server computer – see the networking worksheet where you learnt how to do this. (hint: `ifconfig`)



How it Works

The `import network` line in the above program loads some library code (in this case, it is stored in the `network.py` file provided). This does a lot of the hard work for you in terms of setting up and opening network sockets so that you don't have to bother.

`network.call()` starts the client which connects to another computer via it's IP address. First, we used "localhost" which means "this computer". Later we used the IP address of our other computer.

The client and the server programs both use port 8888, which is the default port used by the network.py module if you don't provide one.

`network.say()` sends a message to the other end of the connection. The connection can be back to your computer ("localhost") or out on to the network to another computer.

<u>That's it! You have now written a network client using a Raspberry Pi!</u>