Installation Snappy:

```
Divyas-MBP:~ divyamereddy$ brew install snappy
Updating Homebrew...
==> Auto-updated Homebrew!
Updated Homebrew from 91f980208 to 646d69613.
Updated 2 taps (homebrew/core, caskroom/cask).
==> New Formulae
ballerina             boost-python3         coreos-ct             howdoi                libbitcoin-consensus  posh                  qtkeychain            shelltestrunner       siril
bareos-client         chrome-export         gocryptfs             krakend               mdcat                 primer3               restview              shogun                unravel
==> Updated Formulae
mercurial ✔                consul                     fonttools                  i2p                        mas                        postgresql@9.5             swiftformat
abnfgen                    cpprestsdk                 fossil                     ideviceinstaller           mbedtls                    pre-commit                 syncthing
advancemame                cryfs                      freedink                   igv                        metabase                   presto                     taisei
akamai                     davmail                    freeipmi                   infer                      metricbeat                 prips                      telegraf
amber                      dbus                       fribidi                    influxdb                   mgba                       proguard                   terraform
angular-cli                dbus-glib                  frugal                     isc-dhcp                   mikutter                   py2cairo                   terragrunt
ansible                    dcmtk                      fuse-zip                   jenkins                    minimal-racket             py3cairo                   tig
```

python:

Strat server:

#!/usr/bin/env python

Licensed to the Apache Software Foundation (ASF) under one

or more contributor license agreements. See the NOTICE file

distributed with this work for additional information

regarding copyright ownership. The ASF licenses this file

to you under the Apache License, Version 2.0 (the

"License"); you may not use this file except in compliance

with the License. You may obtain a copy of the License at

#

http://www.apache.org/licenses/LICENSE-2.0

#

Unless required by applicable law or agreed to in writing, software

distributed under the License is distributed on an "AS IS" BASIS,

```
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
implied.

# See the License for the specific language governing permissions and

# limitations under the License.



#from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer

import http.server

import socketserver

import avro.ipc as ipc

import avro.protocol as protocol

import avro.schema as schema



PROTOCOL = protocol.Parse(open("../avro/mail.avpr").read())



class MailResponder(ipc.Responder):

    def __init__(self):

        ipc.Responder.__init__(self, PROTOCOL)



    def invoke(self, msg, req):

        if msg.name == 'send':

            message = req['message']
```

```
            return ("Sent message to " + message['to']
                    + " from " + message['from']
                    + " with body " + message['body'])
        else:
            raise schema.AvroException("unexpected message:", msg.getname())



Handler = http.server.SimpleHTTPRequestHandler
class MailHandler(Handler):
  def do_POST(self):
    self.responder = MailResponder()
    call_request_reader = ipc.FramedReader(self.rfile)
    call_request = call_request_reader.read_framed_message()
    resp_body = self.responder.respond(call_request)
    self.send_response(200)
    self.send_header('Content-Type', 'avro/binary')
    self.end_headers()
    resp_writer = ipc.FramedWriter(self.wfile)
    resp_writer.write_framed_message(resp_body)
```

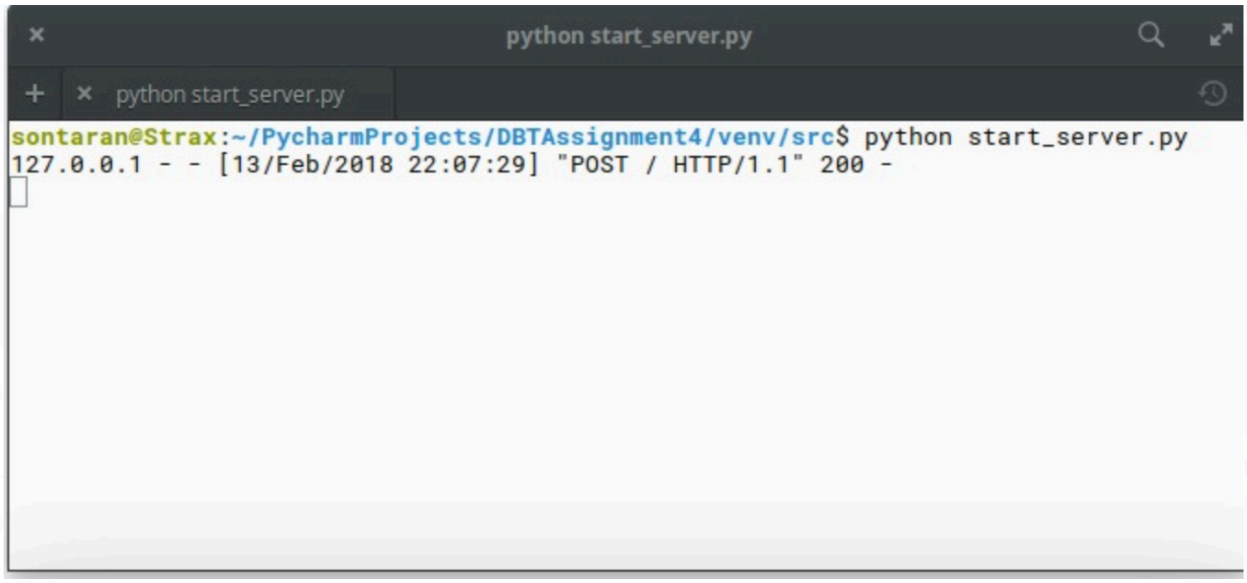
```
server_addr = ('localhost', 9090)

if __name__ == '__main__':

    server = socketserver.TCPServer(server_addr, MailHandler)

    server.allow_reuse_address = True

    server.serve_forever()
```



Run hello world:

```
import sys

import http.client as client

import avro.ipc as ipc
import avro.protocol as protocol

PROTOCOL = protocol.Parse(open("../avro/mail.avpr").read())

server_addr = ('localhost', 9090)

class UsageError(Exception):
   def __init__(self, value):
```

```
        self.value = value
    def __str__(self):
        return repr(self.value)

if __name__ == '__main__':
    if len(sys.argv) != 4:
        raise UsageError('Usage: <to> <from> <body>')

    # client code - attach to the server and send a message
    client = ipc.HTTPTransceiver(server_addr[0], server_addr[1])
    requestor = ipc.Requestor(PROTOCOL, client)

    # fill in the Message record and send it
    message = dict()
    message['to'] = sys.argv[1]
    message['from'] = sys.argv[2]
    message['body'] = sys.argv[3]

    params = dict()
    params['message'] = message
    print ("Result: " + requestor.request('send', params))

    # cleanup
    client.close()
```