



# 国連ベクトルタイル ツールキット 技術移転教材「わし」

2020-08

国連ベクトルタイルツールキット

# Scope and audience

- Scope of this document

- オープンに提供されている地理空間情報からベクトルタイルを生産して、ウェブ地図サーバを動かす方法を記述する。
- ベクトルタイルのスタイルを変更する方法を記述する。
- ベクトルタイルのサイズについて指標を算出し、ベクトルタイルのサイズを最適化するための方法を記述する。

- このドキュメントの読者

- ベクトルタイルを生産して、ウェブ地図サーバを動かすための方法を身につけたい方。
- 地理空間情報にある程度親しんでいる方。
- コンピュータの取り扱いについては、詳しい知識はなくても、新しい操作に興味を持てる方。

# Executive Summary

- (あとで書く)

- 
- (なぜ、あなた方と私たちはともにベクトルタイルを生産・ホスト・設計・最適化する必要があるのか、をかんたんに、非技術的に書く)
  - 私たちは、地図のプラットフォームのその下、地図のインフラストラクチャーを提供してきた。デジタルの前からそうしていたからだ。
  - 私たちは、インフラストラクチャーはブラックボックスではいけないと思った。費用対効果だけではなく、相互運用性とスケーラビリティのために。
  - 私たちは、それで地理空間情報当局のウェブ地図を運営している。その運営実績を高く評価されて国連にも招聘され2年を過ごした。
  - そうして生まれた国連ベクトルタイルツールキットで、先進的なウェブ地図をともにつくろう。

- ここでは、小さいけど本物のウェブ地図を作る。
- 本物というのは、強靭でスケーラブルということだ。
- 強靭なウェブのプラットフォームというのは、クラウドサービスなどで特にそうなのだが、Unix で動いている。
- なので、UNVT も Unix に載っている。
- 難しいところはそこかもしれない。でも、そこと一緒に進もう。
- スケーラブルというのは、大きなデータでも、たくさんのアクセスを受けても対応可能ということだ。
- UNVT は、全世界の OpenStreetMap データを 80 時間でベクトルタイルに変換できる。
- 一国のデータであれば、1日から数日で変換できるはずだ。

# スタートとゴール

## スタート

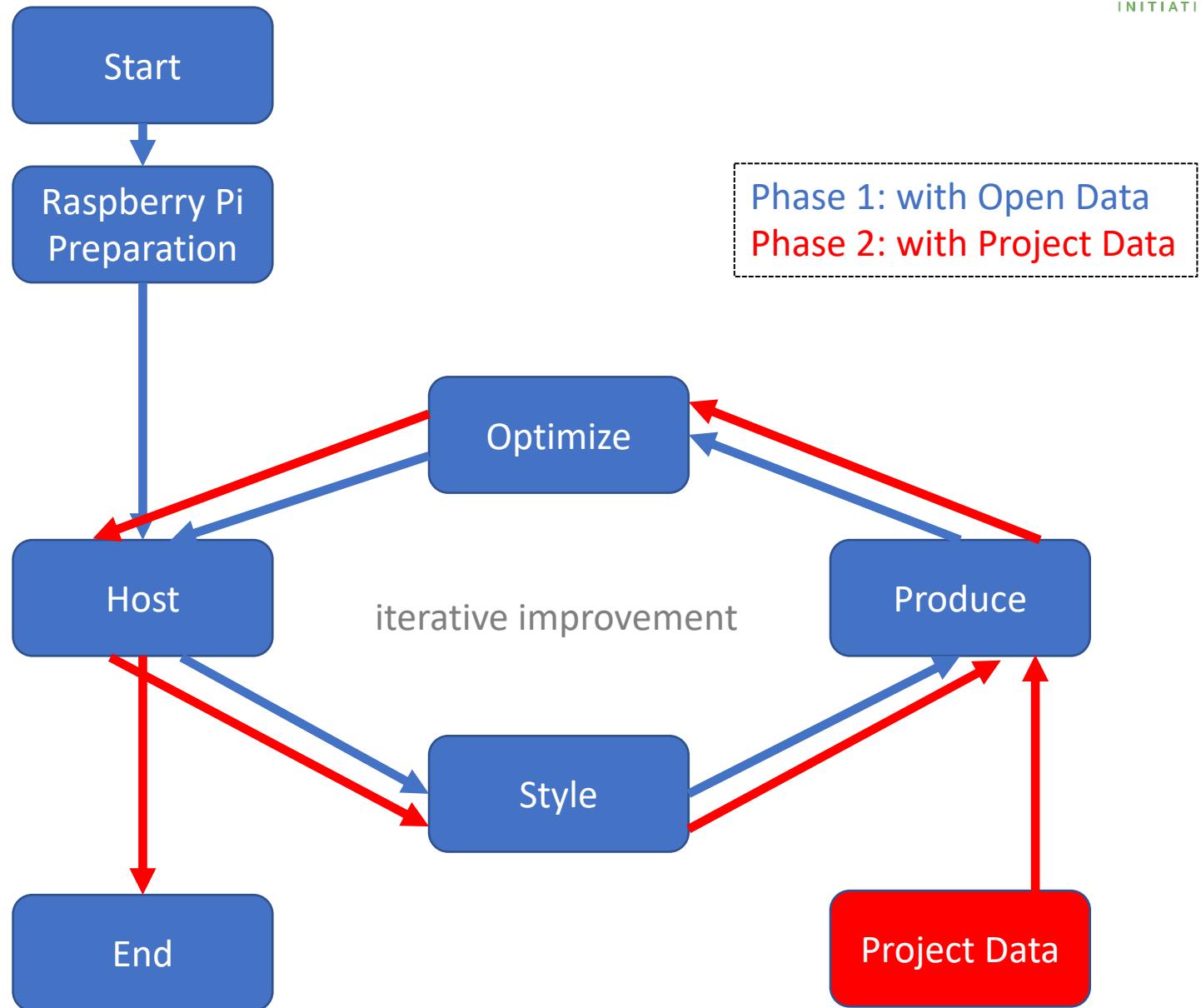
- ✓ ハードウェア
  - ✓ PC
  - ✓ Raspberry Pi 4
  - ✓ LAN ケーブル
- ✓ インターネット接続
  - ✓ インターネットへの接続
  - ✓ WiFi
  - ✓ Ethenet 接続
    - ✓ Raspberry Pi の初期設定用



## ゴール

- ✓ Tirana-Durres Vector Tiles: アルバニアのティラナ・ドゥレス地域の先進的なウェブ地図

# Flow



# この技術移転ドキュメントについて

8



- 難易度の高いスライドには *ninja* がついています。*ninja* がついたスライドは、分からなくても先に進んで大丈夫です。



- 理由を説明するスライドには *questioning woman* がついています。



- Raspberry Pi を開封して部品を理解する。

- ケースがあれば入れる。

- PC で Raspberry Pi OS をダウンロードして Micro SD カードに書き込む。

- Ethernet ケーブルで PC と Raspberry Pi をつなぐ
- Raspberry Pi に電源を接続して、立ち上げる。

# ssh pi@raspberrypi.local

- default password: raspberrypi

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
[pc:washi $ ssh pi@raspberrypi.local
pi@raspberrypi.local's password:
Linux raspberrypi 5.4.51-v8+ #1327 SMP PREEMPT Thu Jul 23 11:11:34 BST 2020 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug 21 15:40:29 2020 from fe80::25:ead:c69:349d%wlan0
pi@raspberrypi:~ $ ]
```

# sudo apt update

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
individual files in /usr/share/doc/*copyright.

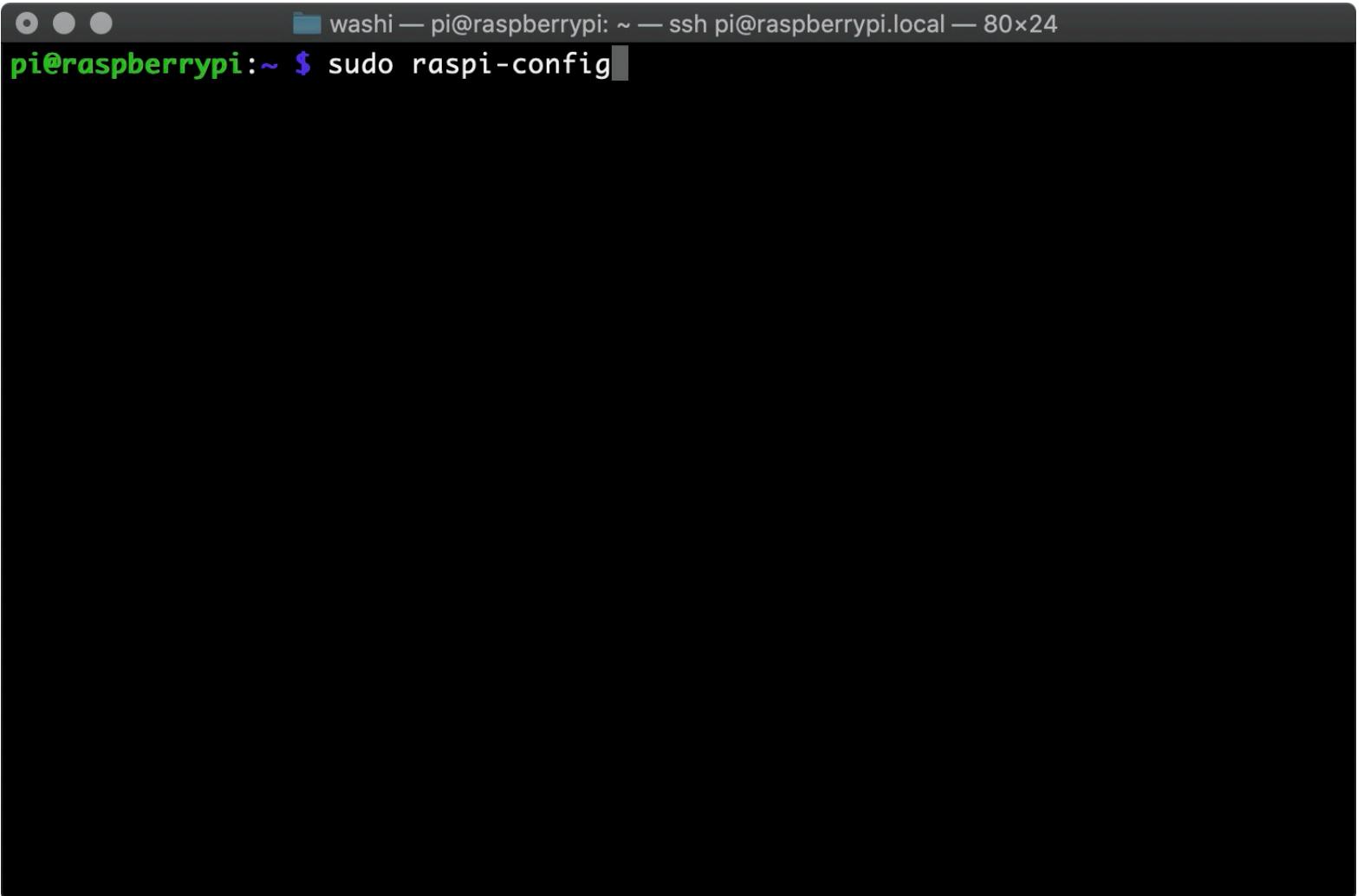
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug 21 15:40:29 2020 from fe80::25:ead:c69:349d%wlan0
[pi@raspberrypi:~ $ sudo apt update
Hit:1 https://dl.yarnpkg.com/debian stable InRelease
Hit:2 https://deb.nodesource.com/node_14.x buster InRelease
Hit:3 http://deb.debian.org/debian buster InRelease
Get:4 http://deb.debian.org/debian-security buster/updates InRelease [65.4 kB]
Get:5 http://deb.debian.org/debian buster-updates InRelease [51.9 kB]
Hit:6 http://archive.raspberrypi.org/debian buster InRelease
Get:7 http://deb.debian.org/debian-security buster/updates/main armhf Packages [216 kB]
Get:8 http://deb.debian.org/debian-security buster/updates/main arm64 Packages [216 kB]
Get:9 http://deb.debian.org/debian-security buster/updates/main Translation-en [120 kB]
Fetched 668 kB in 2s (412 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
pi@raspberrypi:~ $ ]
```

# sudo apt upgrade

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
[pi@raspberrypi:~ $ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
pi@raspberrypi:~ $ ]
```

# sudo raspi-config

raspi-config で WiFi の設定を行う。

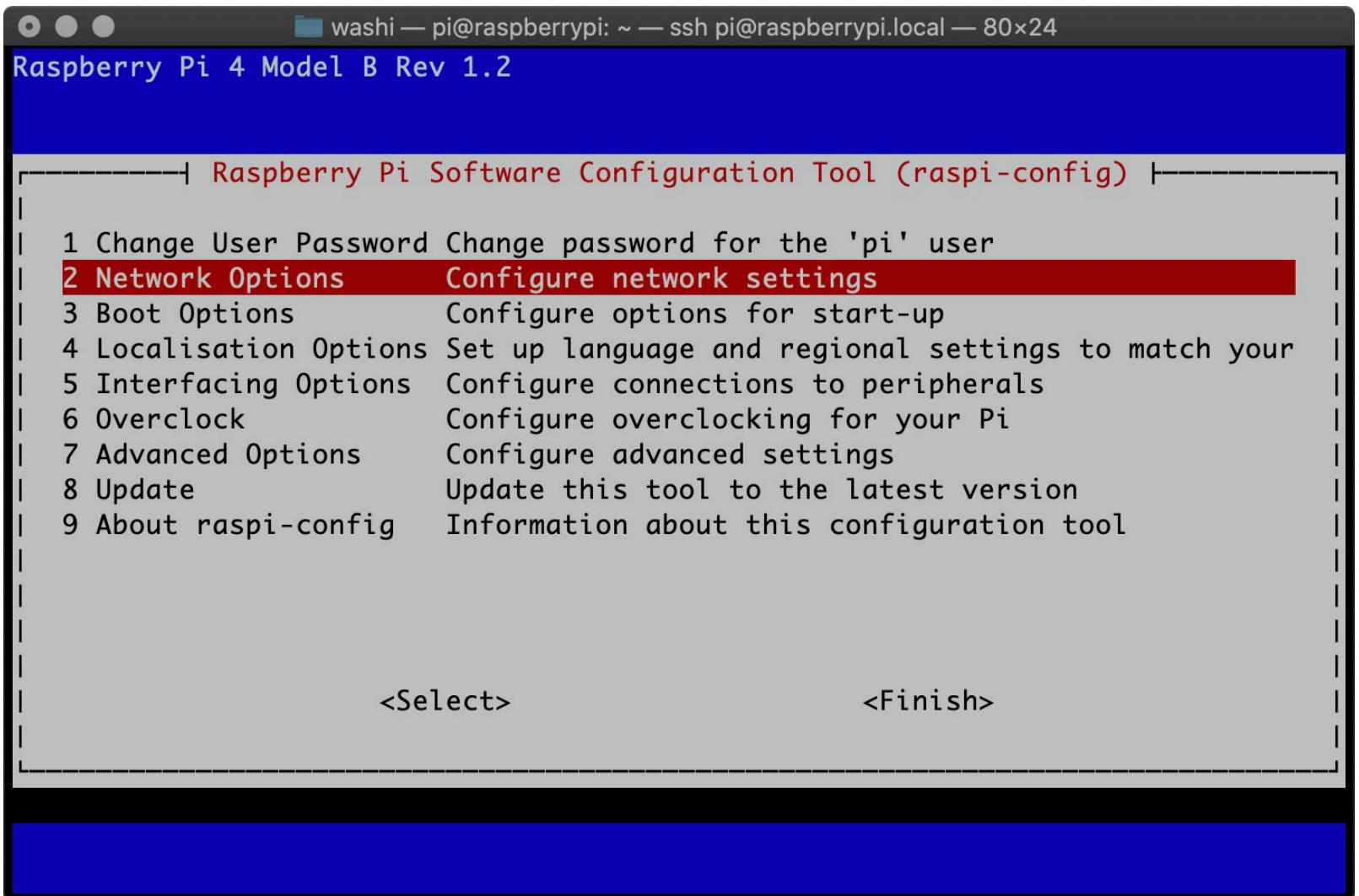


washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24

```
pi@raspberrypi:~ $ sudo raspi-config
```

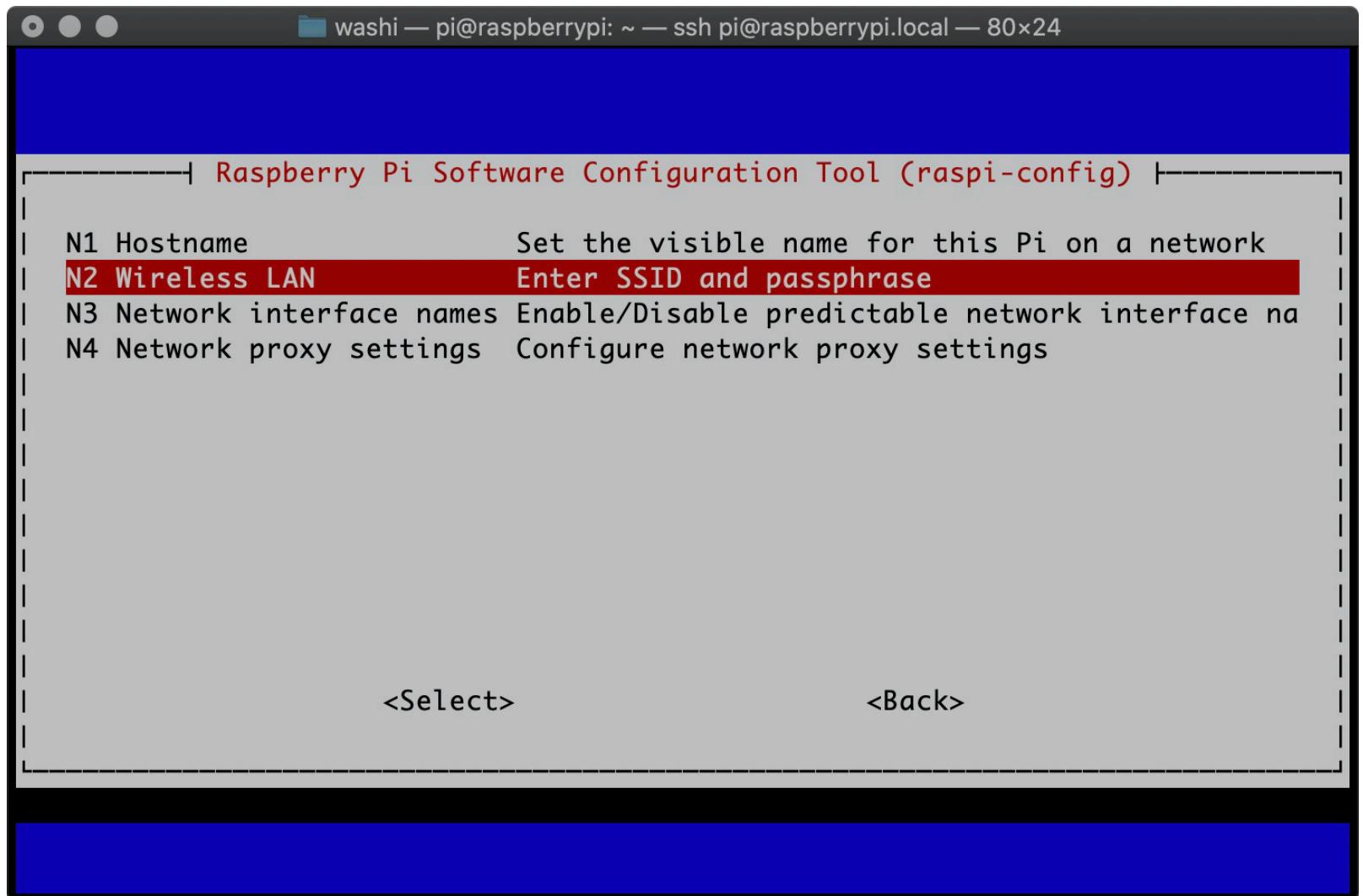
# 2 Network Options

raspi-config で WiFi の設定を行う。



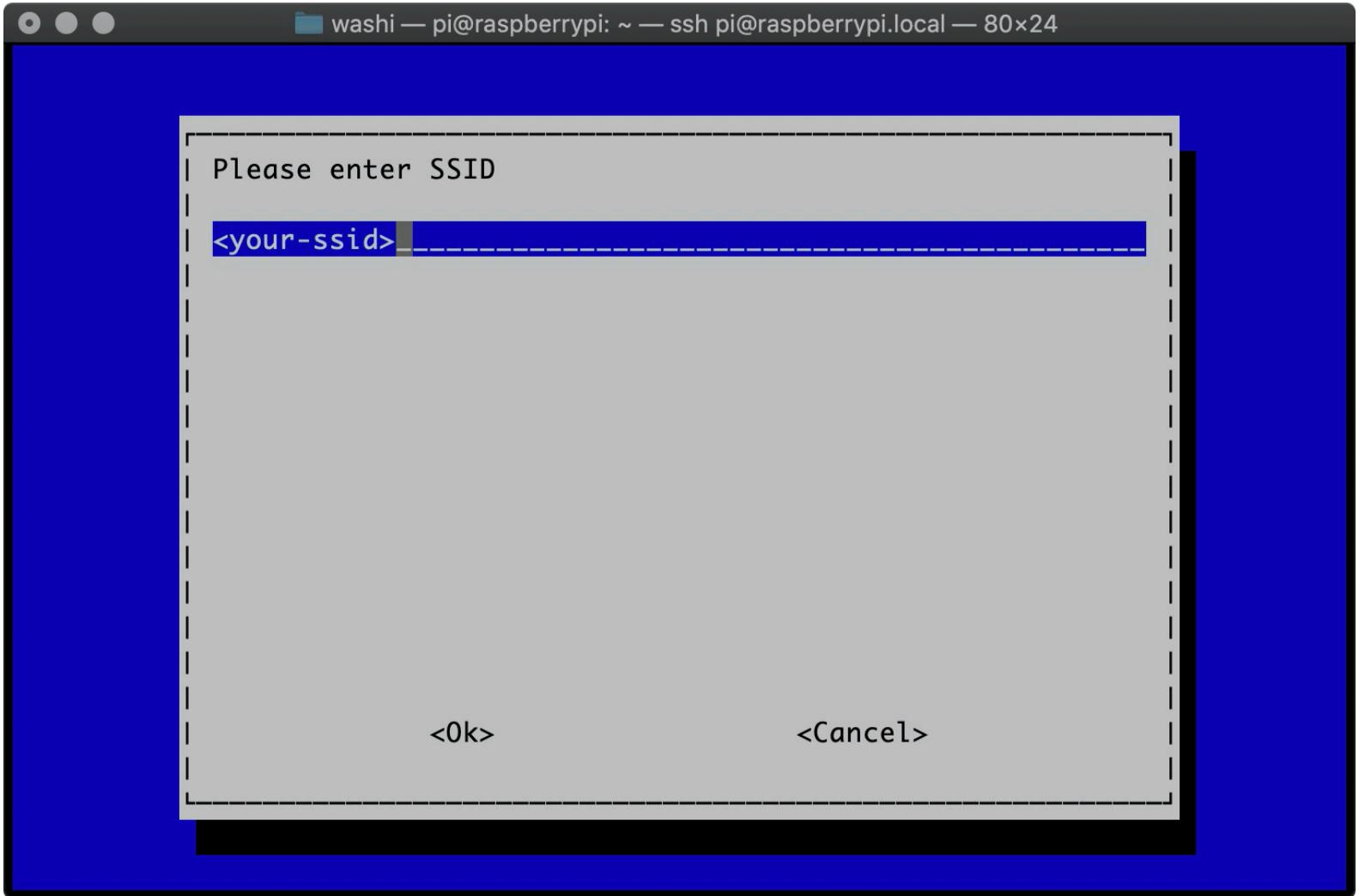
# N2 Wireless LAN

raspi-config で WiFi の設定を行う。



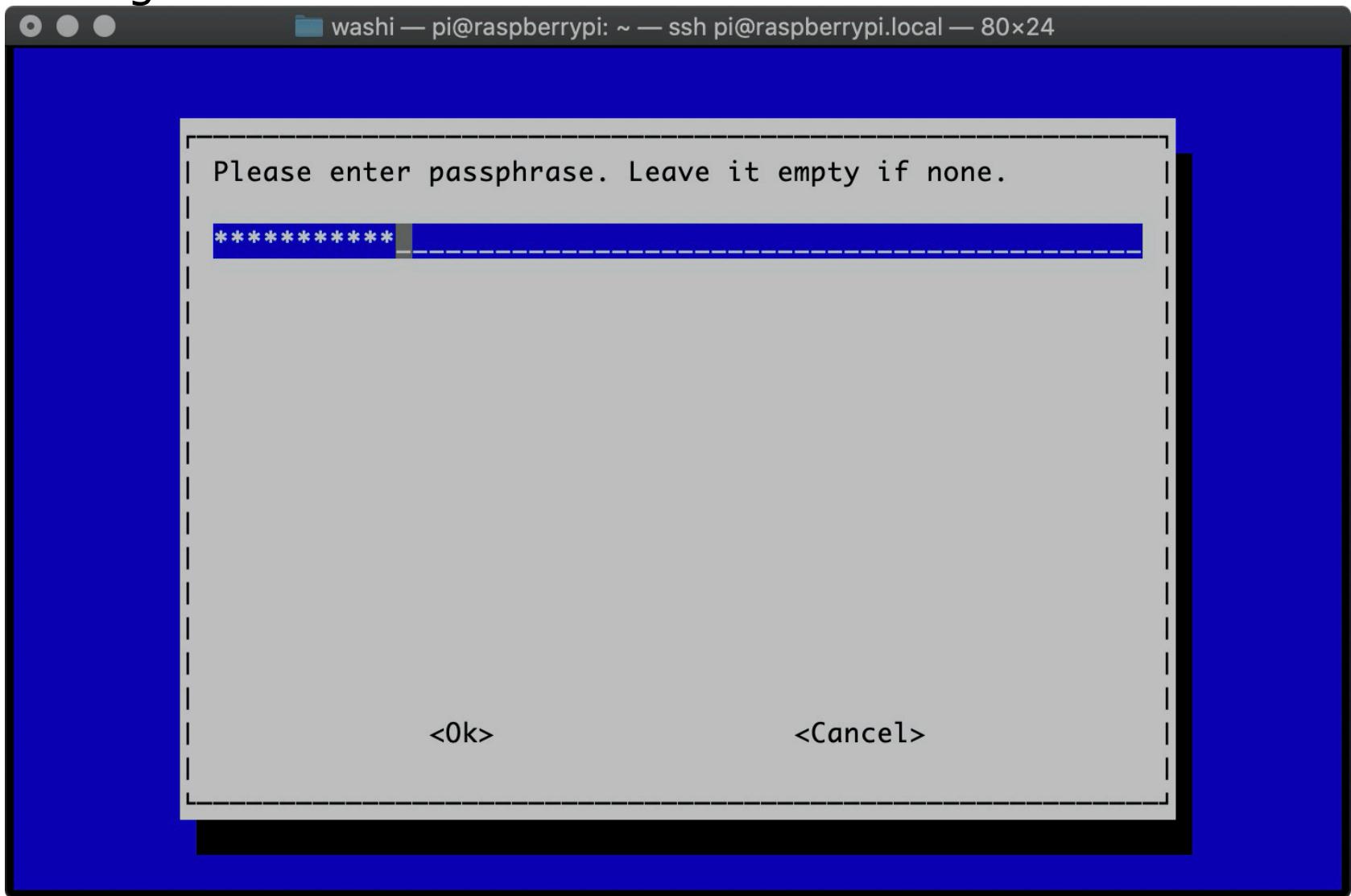
# Enter your SSID

raspi-config で WiFi の設定を行う。



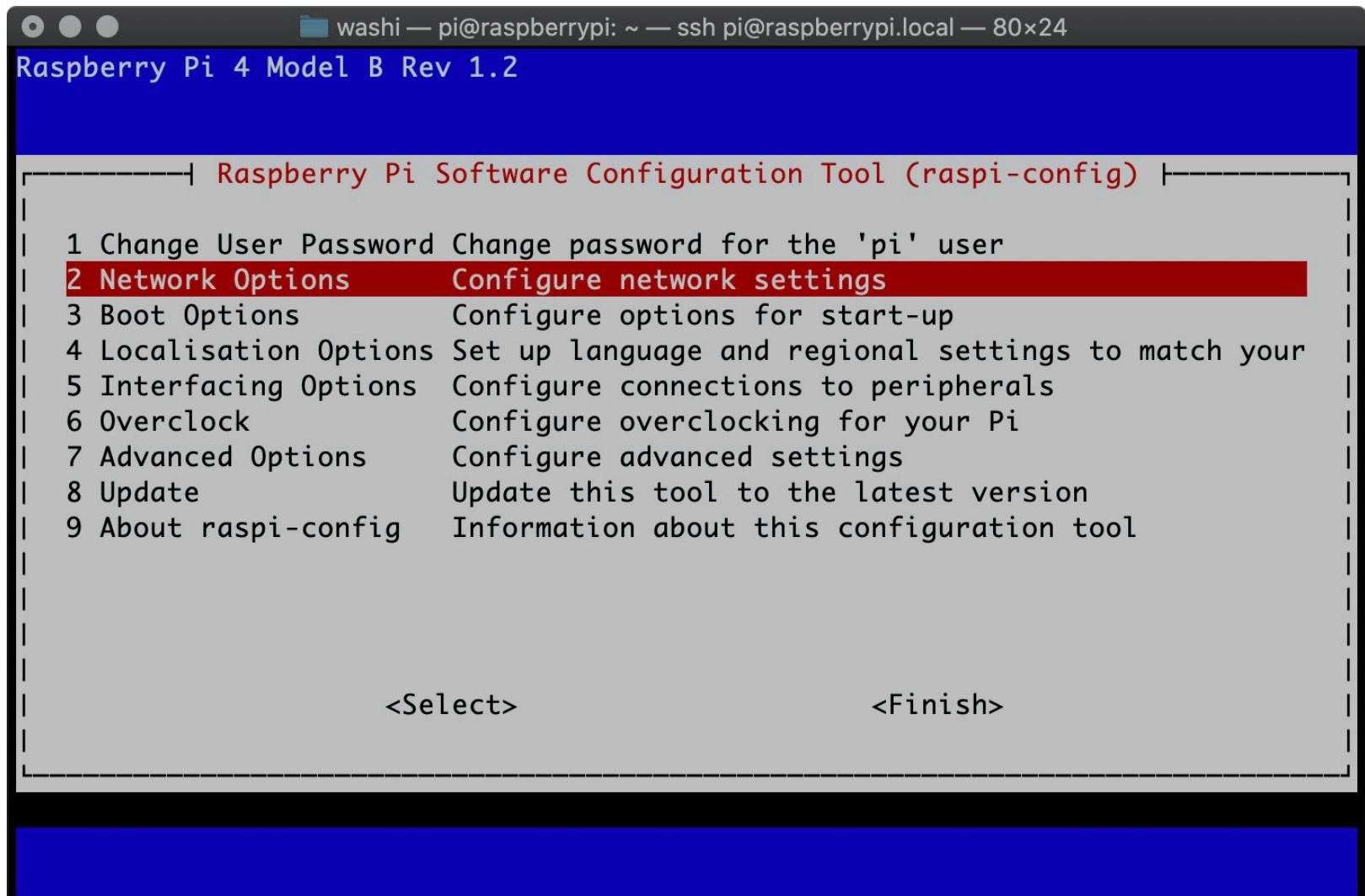
# Enter your WiFi passphrase

raspi-config で WiFi の設定を行う。



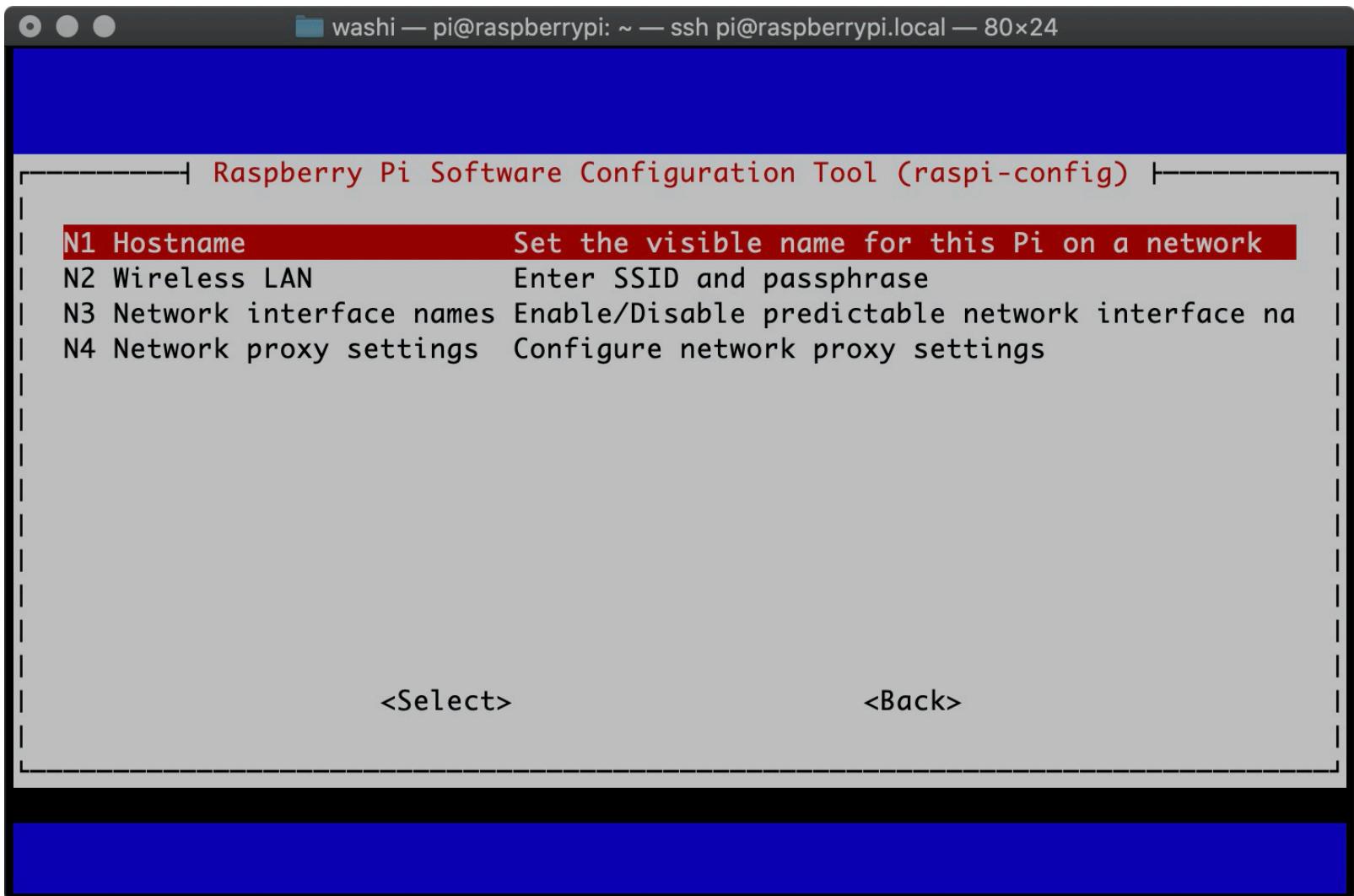
# 2 Network Options

Change your hostname



# N1 Hostname

Change your hostname

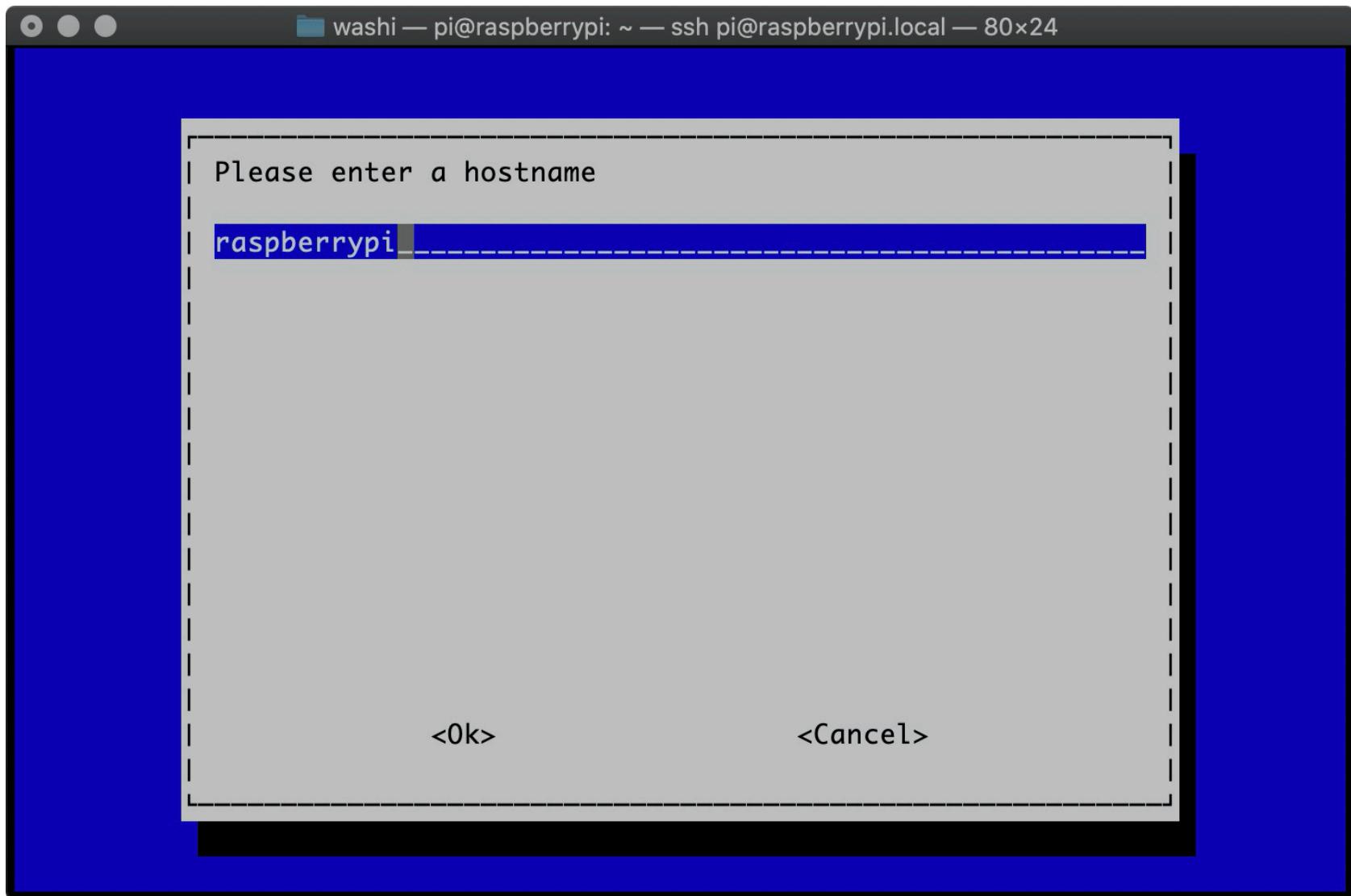


```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24

| Please note: RFCs mandate that a hostname's labels may contain
| only the ASCII letters 'a' through 'z' (case-insensitive),
| the digits '0' through '9', and the hyphen.
| Hostname labels cannot begin or end with a hyphen.
| No other symbols, punctuation characters, or blank spaces are
| permitted.

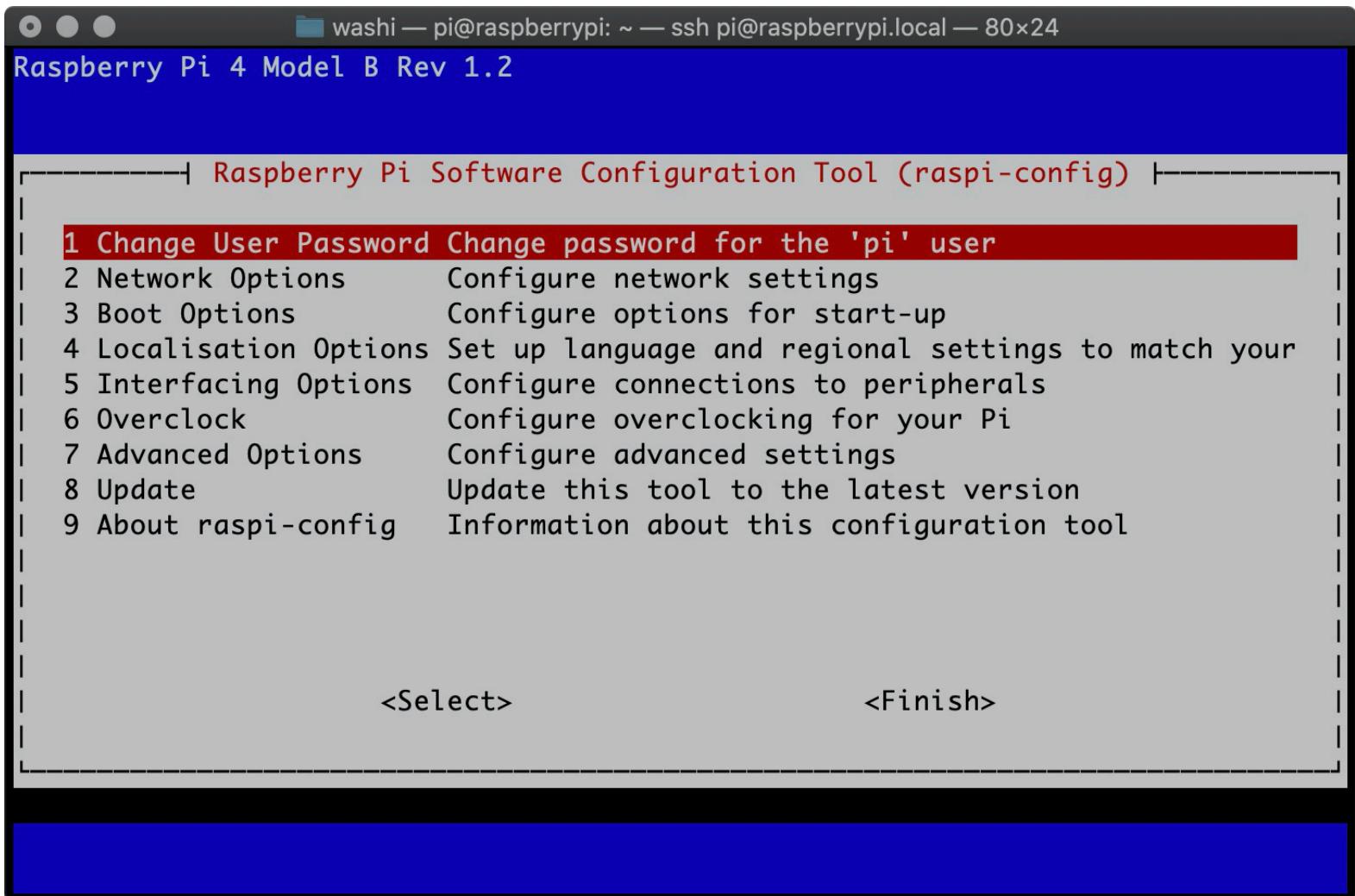
| <0k>
```

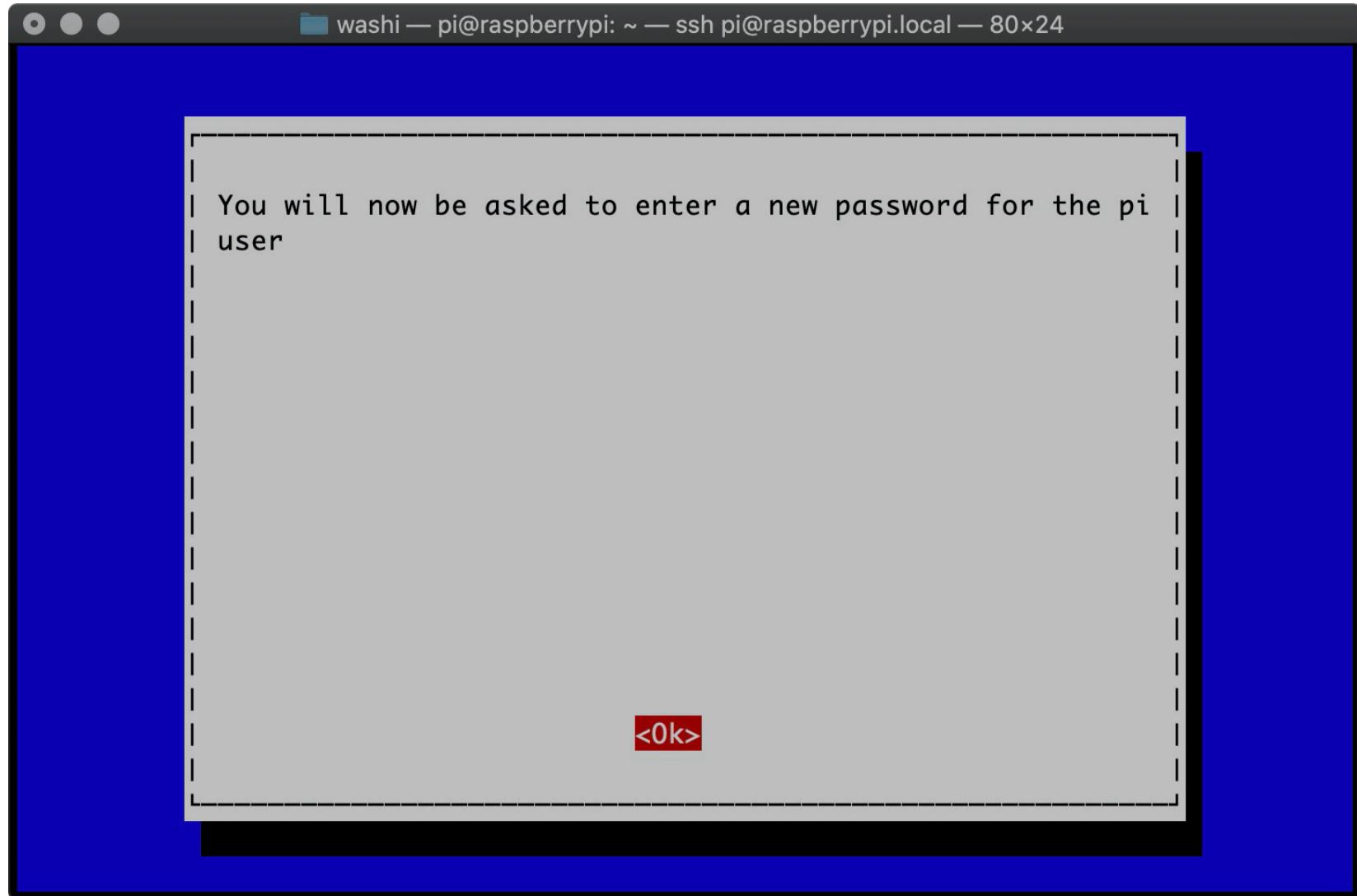
# Enter your new hostname



# Change password for pi

- ・パスワードも変更する。（ユーザについては、複雑さを避けるために pi のままということで説明をする。）



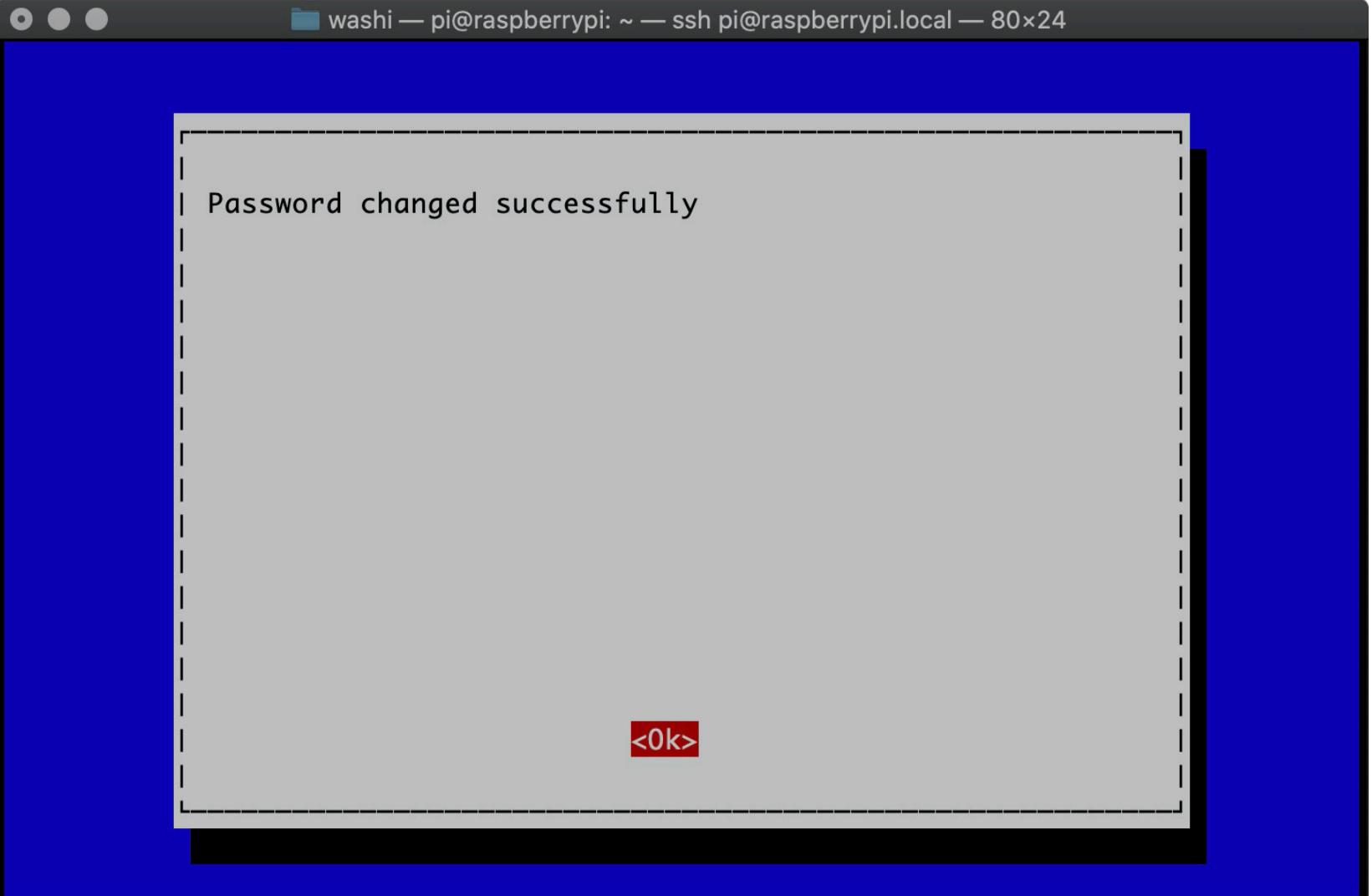


washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24

```
You will now be asked to enter a new password for the pi user

<0k>
```

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
[pi@raspberrypi:~ $ sudo raspi-config
New password:
Retype new password:
```



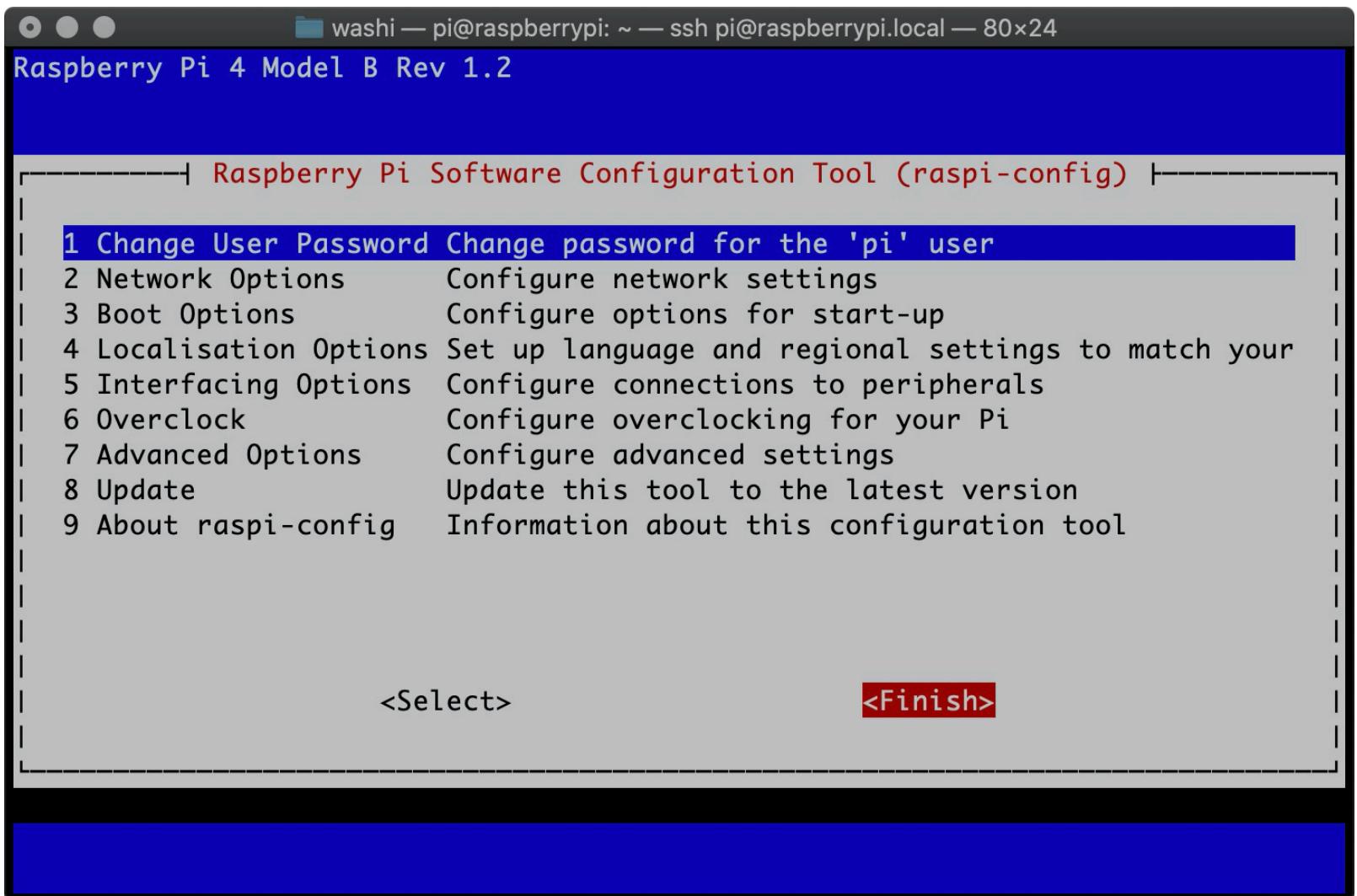
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24

```
 Password changed successfully
<0k>
```

The terminal window shows a message "Password changed successfully" in white text on a black background. At the bottom right, there is a red button-like element containing the text "<0k>" in white. The terminal title bar indicates the session is on a Raspberry Pi with user "washi" and host "pi@raspberrypi.local". The window has a standard OS X-style title bar with three circular buttons.

# Finish

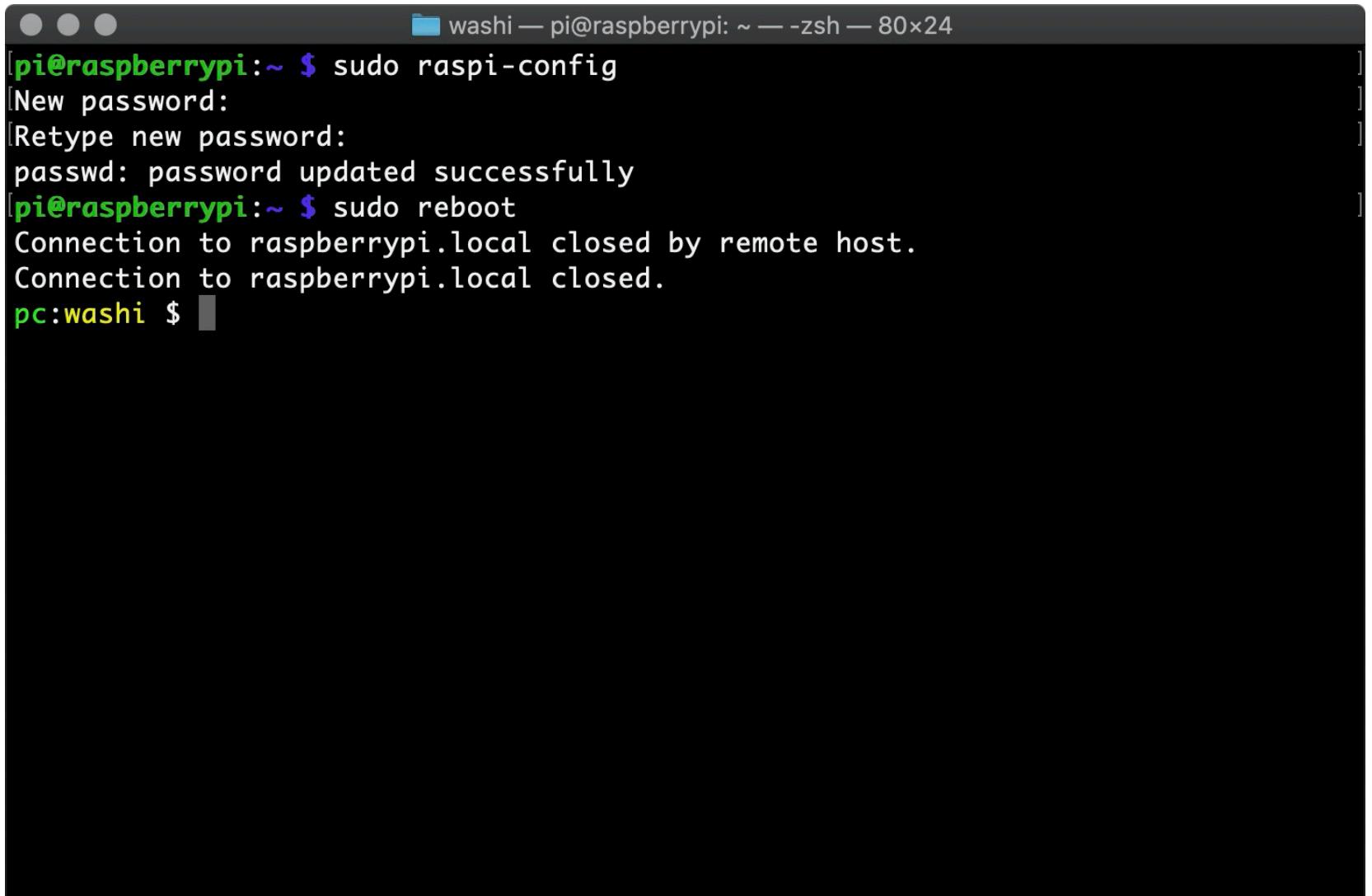
- Use tab key until <Finish> turns to red. Then enter.



- Raspberry Pi を再起動し、PC から WiFi 経由でログインをする。

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
[pi@raspberrypi:~ $ sudo raspi-config
[New password:
[Retype new password:
passwd: password updated successfully
pi@raspberrypi:~ $ sudo reboot]
```

- Raspberry Pi を再起動し、PC から WiFi 経由でログインをする。



```
washi — pi@raspberrypi: ~ — -zsh — 80x24
[pi@raspberrypi:~ $ sudo raspi-config
[New password:
[Retype new password:
passwd: password updated successfully
[pi@raspberrypi:~ $ sudo reboot
Connection to raspberrypi.local closed by remote host.
Connection to raspberrypi.local closed.
pc:washi $ ]
```

- Raspberry Pi を再起動し、PC から WiFi 経由でログインをする。

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
[pi@raspberrypi:~ $ sudo raspi-config
[New password:
[Retype new password:
passwd: password updated successfully
[pi@raspberrypi:~ $ sudo reboot
Connection to raspberrypi.local closed by remote host.
Connection to raspberrypi.local closed.
[pc:washi $ ssh pi@raspberrypi.local
pi@raspberrypi.local's password: ?]
```

```
washi — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
[pi@raspberrypi:~ $ sudo raspi-config
>New password:
Retype new password:
passwd: password updated successfully
[pi@raspberrypi:~ $ sudo reboot
Connection to raspberrypi.local closed by remote host.
Connection to raspberrypi.local closed.
[pc:washi $ ssh pi@raspberrypi.local
pi@raspberrypi.local's password:
Linux raspberrypi 5.4.51-v8+ #1333 SMP PREEMPT Mon Aug 10 16:58:35 BST 2020 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Aug 24 12:55:43 2020 from fe80::25:ead:c69:349d%wlan0
pi@raspberrypi:~ $ ]
```



34



- Unix とは何か。 CLI とは何か。 [linuxjourney.com](http://linuxjourney.com), [vimtutor](http://vimtutor.org)

- equinox をインストールする。

- 各コマンドの version 又は help 情報が出ることを確認する。

- GitHub のユーザーアカウントを持っていなければ、ここで作る。

- optgeo/sh2 を fork する



- GitHub の鍵を Raspberry Pi にセットする。

- レポジトリ \${you}/sh2 を clone する



# host

- (localhost で) ホストしてみる
- cd sh2
- yarn add
- rake host
- access [https://\(your host\):port](https://(your host):port)



- rake / Rakefile について





44



- ・ブラウザのデベロッパーツールでデータの動きを確認してみる



45



- 座学的に、ウェブ地図の原理を解説する。

[English](#) / [Italiano](#) / [Español](#) / [Svenska](#) / [Français](#) / [日本語](#) / [Português](#) / [Deutsch](#) / [Dutch](#) / [Ukrainian](#) / [한국어](#)

## mapschool

a free introduction to geo

[home](#)   [code](#)   [feedback](#)   [contributors](#)   [further reading](#)

### map school

What is a map? Until the 1980s, maps were painstaking documents created by hand. These days maps are almost always made with the help of a computer. Maps today are commonplace, interspersed in driving directions, visualizations, and political boundary disputes. Let's look deeper and think about the fundamental elements of maps from the eye of the creator.

Computer maps are fundamentally composed of data. Data is in the abstract, composed of billions of points or just a few polygons, or a photo-like recording of colors and temperature. It is important that data is not specific to a certain usage.

From data, we make numbers, pictures, and decisions. Most commonly we make

# How we got here

Tom MacWright  
@tmcw / @Mapbox



46



- localhost.run 又は serveo を使って、インターネットからアクセス可能にしてみる



47



- gh-pages でホストする方法を紹介する

- <宿題：テキストエディタ vi か nano を使えるようになってきてください。>

# style

- Mapbox Style の概要
- style.json を見てみる

- `parse-hocon` と `gl-style-validate` を使い、複雑なスタイルを心理的に安全に取り扱う。

- 実際にスタイルを変更してみる。
- テキストエディタで編集して、rake style, rake host

# produce



54



- ・<座学説明>ベクトルタイルとは、ベクトルデータから生産されたタイル。
- ・ベクトルデータを復元可能である。
- ・ベクトルタイルはベクトルデータへの窓である。
- ・ベクトルタイルはベクトルデータの空間的なストリーミング方法である。

- ・<座学説明> Tippecanoe はタイル生産を行うツール。
- ・ここでは、Tippecanoe に tippecanoe 属性を加えた GeoJSONs をパイプで渡す。
- ・出力は mbtiles というパッケージファイルである。tile-join というツールを使ってファイルシステムに展開する必要がある。

- ・<座学説明> 今回は OpenStreetMap の Tirana-Durres 地域を切り出してみる。
- ・osmium というツールを使って、切り出しを行う。

- geofabrik から、OpenStreetMap のアルバニア extract をダウンロードする。
- curl というツールを使って、ダウンロードを行う。

- rake dump で見てみる。
- Rakefile を見ると、確かに地域限定して切り出しをしている。



59



- <座学説明> stream と pipe
- データをストレージしないで後段に渡すので、無駄が少ない。

- 「| node filter.js」を加えて、tippecanoe 属性が加わったことを確認する。

- rake tiles
  - Tippecanoe で変換をする
  - tile-join で展開をする

- rake host してみて、確かにウェブ地図が動いていることを確認する

- filter.js を書き換えて、building が通らないようにしてみる。

- rake tiles
  - Tippecanoe で変換する
  - tile-join で展開をする

- rake host してみて、確かにウェブ地図が動いていることを確認する

# optimize



67



- ・<座学説明>どのような考え方で filter.js を設計していくか。
- ・タイルのサイズを現実的な大きさにすることが重要である。
- ・ベクトルタイルは人間の目が見るデータであるから、必ずサイズと情報内容を両立する設定がある。
- ・機械が見るベクトルタイルであっても、オーダーは変わるかもしれないにせよ、同じかもしれない。

- 指標に基づく最適化が大事。
- vt-optimizer を動かしてみる。

- filter.js を書き換えて、建物が再び含まれるようにする。

- rake tiles; rake host して、ウェブ地図としてワークすることを確かめる。

# まとめ

- まとめ

# 次に行うこと

73



- ・プロジェクトのデータを用いた、インターネットからアクセスできる、ベクトルタイルウェブ地図をつくる