

```
    return data[query.idx]
  } else if (fn.endsWith('/clear')) {
    data = []
    save()
    return data
  } else if (fn.endsWith('/reset')) {
    data.splice(query.idx, 1)
    save()
    return data
  }
}
return { res: "OK" }
}

const server = http.createServer()
server.on('request', function (req, res) {
  console.log(req.url)
  if (req.url.startsWith('/api')) {
    const urlp = url.parse(req.url, true)
    res.writeHead(200, { 'Content-Type': 'application/json' })
    const resjson = serveAPI(urlp)
    res.write(JSON.stringify(resjson))
    res.end()
  } else {
    simplewebserver.serve(res)
  }
})
server.listen(8001)
```

Nodeでチャット！

たいすけ 発言いれて 発言！

1584312437414: たいすけ: はろー

一番上、ひとつ消す 全部消す！

Nodeでチャット！

localhost:8001/chat.html

Nodeでチャット！

localhost:8001/chat.html

ふくの 発言いれて 発言！

1584312437414: たいすけ: はろー

1584312447731: ふくの: こんにちは！

一番上、ひとつ消す 全部消す！

Webサービスはじめてのいっぽ JavaScript / Node.js版



CC BY @taisukef <https://fukuno.jig.jp/2790>

コンピューターと
もっと深く話そう



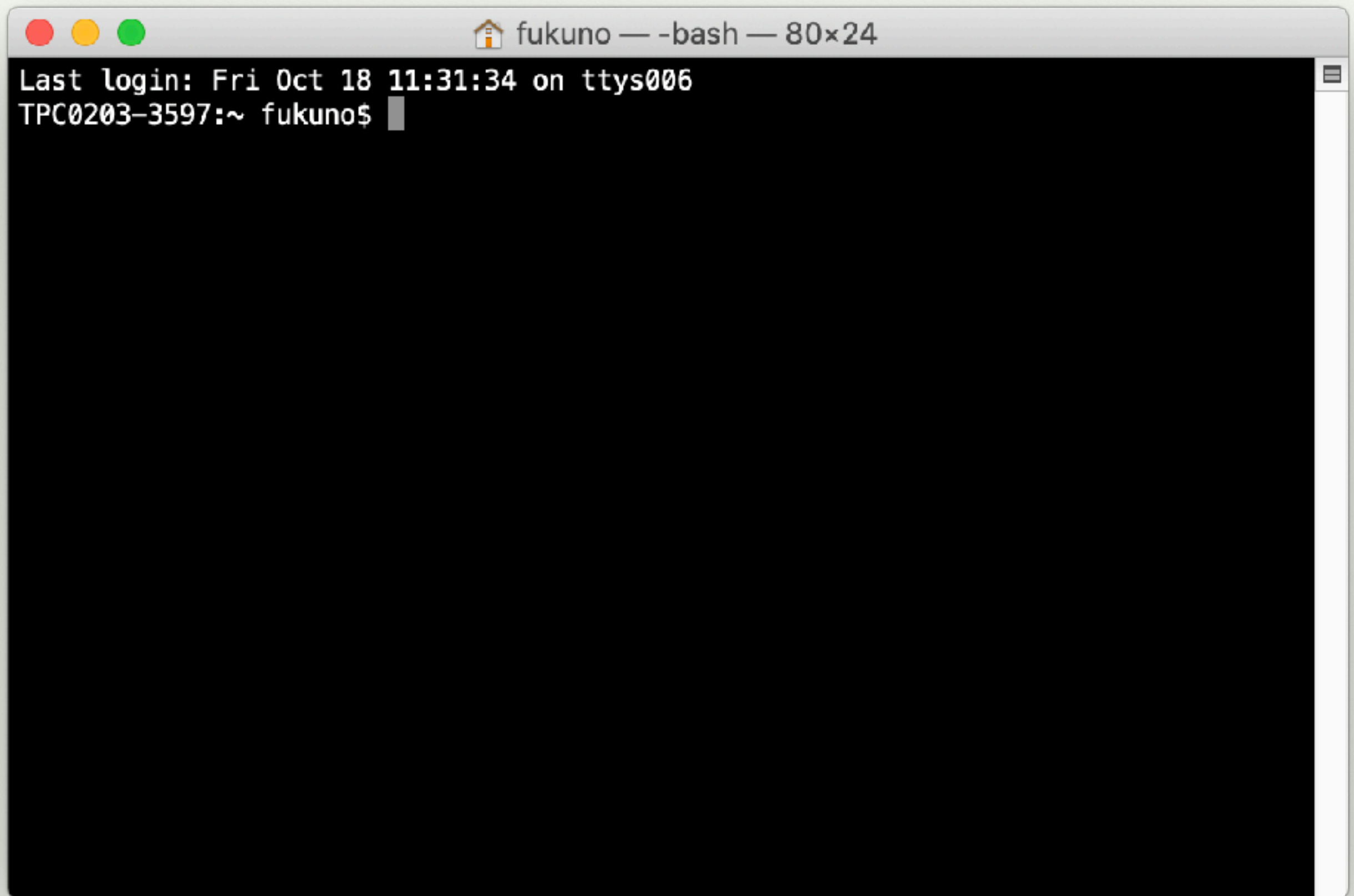


...

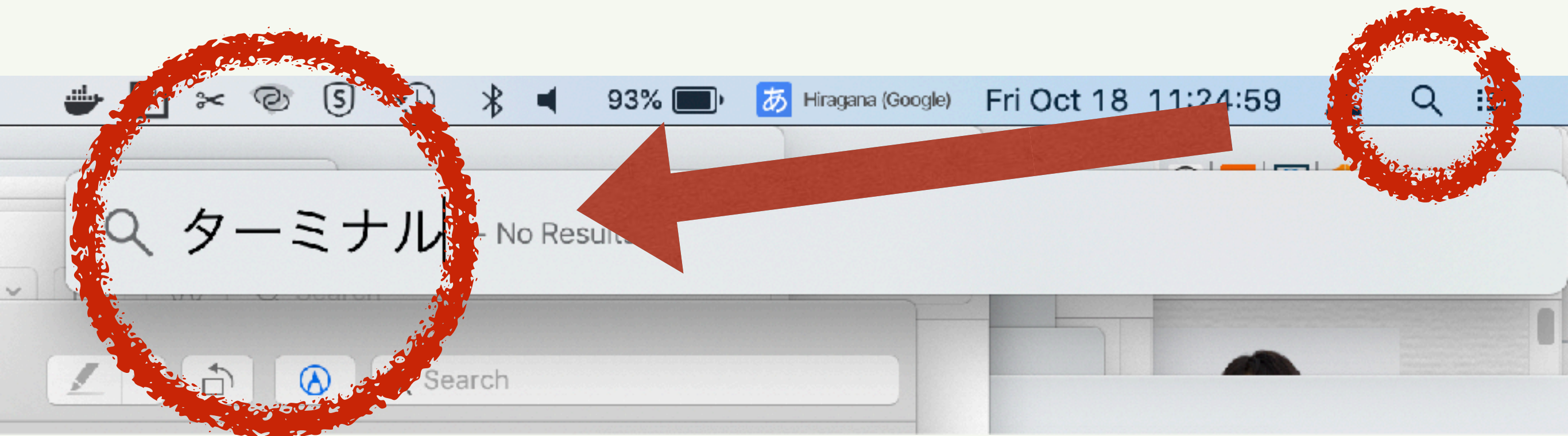
(キイテナイヨ)

ハロー





ターミナル！



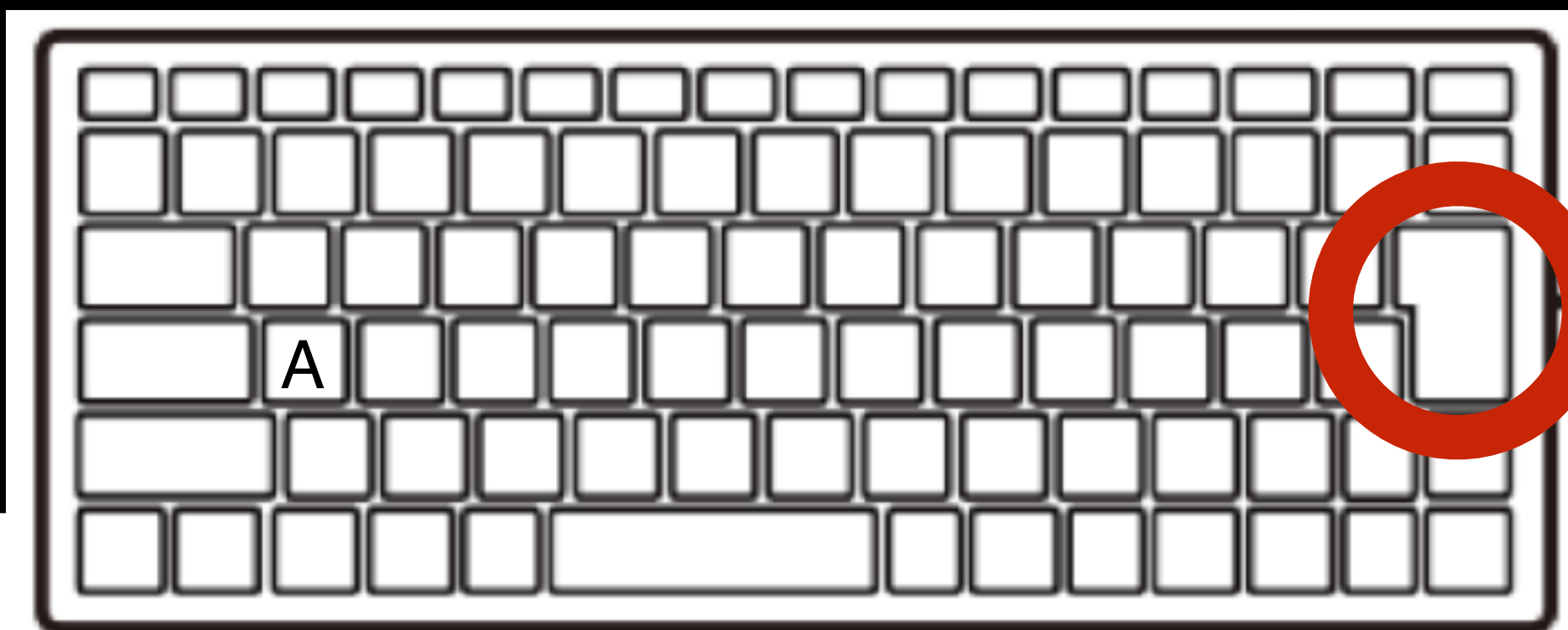
「ターミナル」を開く

* Windowsの場合: Windowsキー+R → 「cmd」 でエンター

```
Last login: xxx  
$a|
```

キーボードで「a」と、うってみよう

```
Last login: xxx  
$a|
```



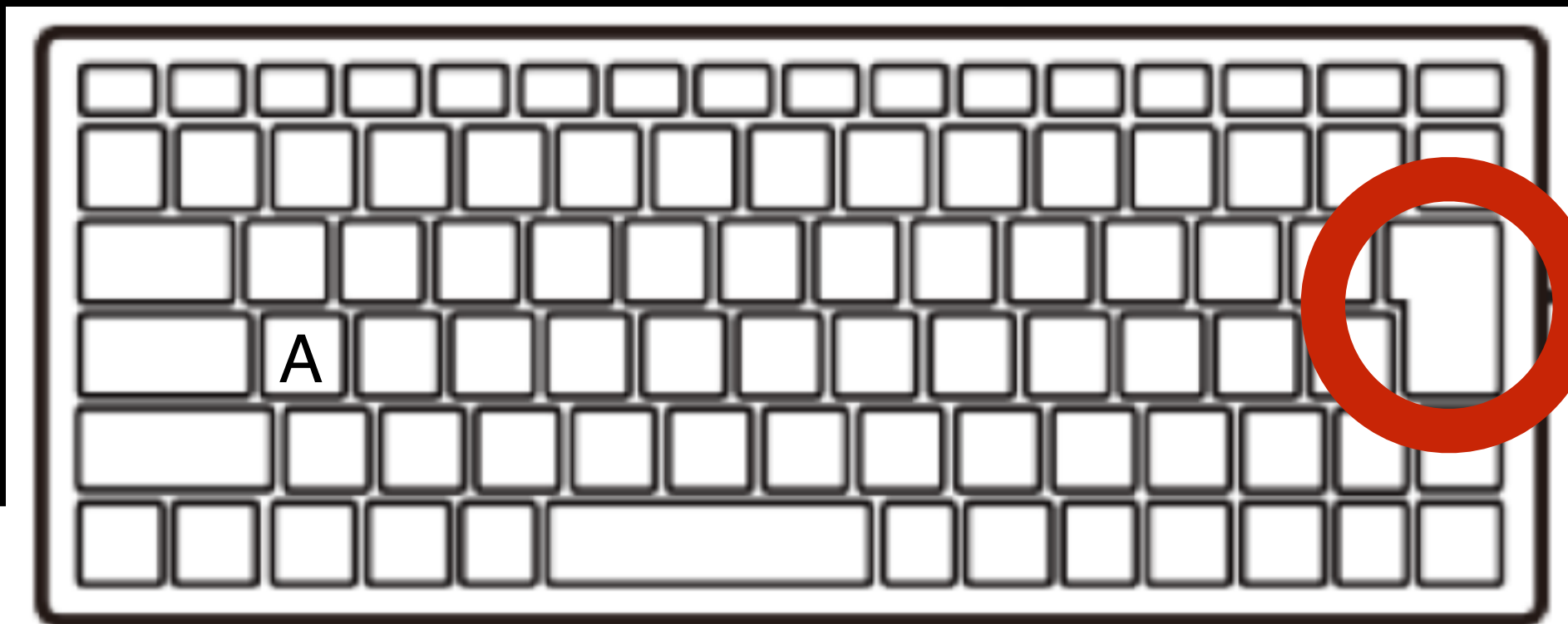
エンターキー

```
Last login: xxx
```

```
$a
```

```
-bash: a: command not found
```

```
$
```



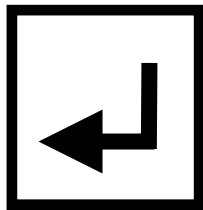
エンターキー

?



シラナイ
コトバダナー

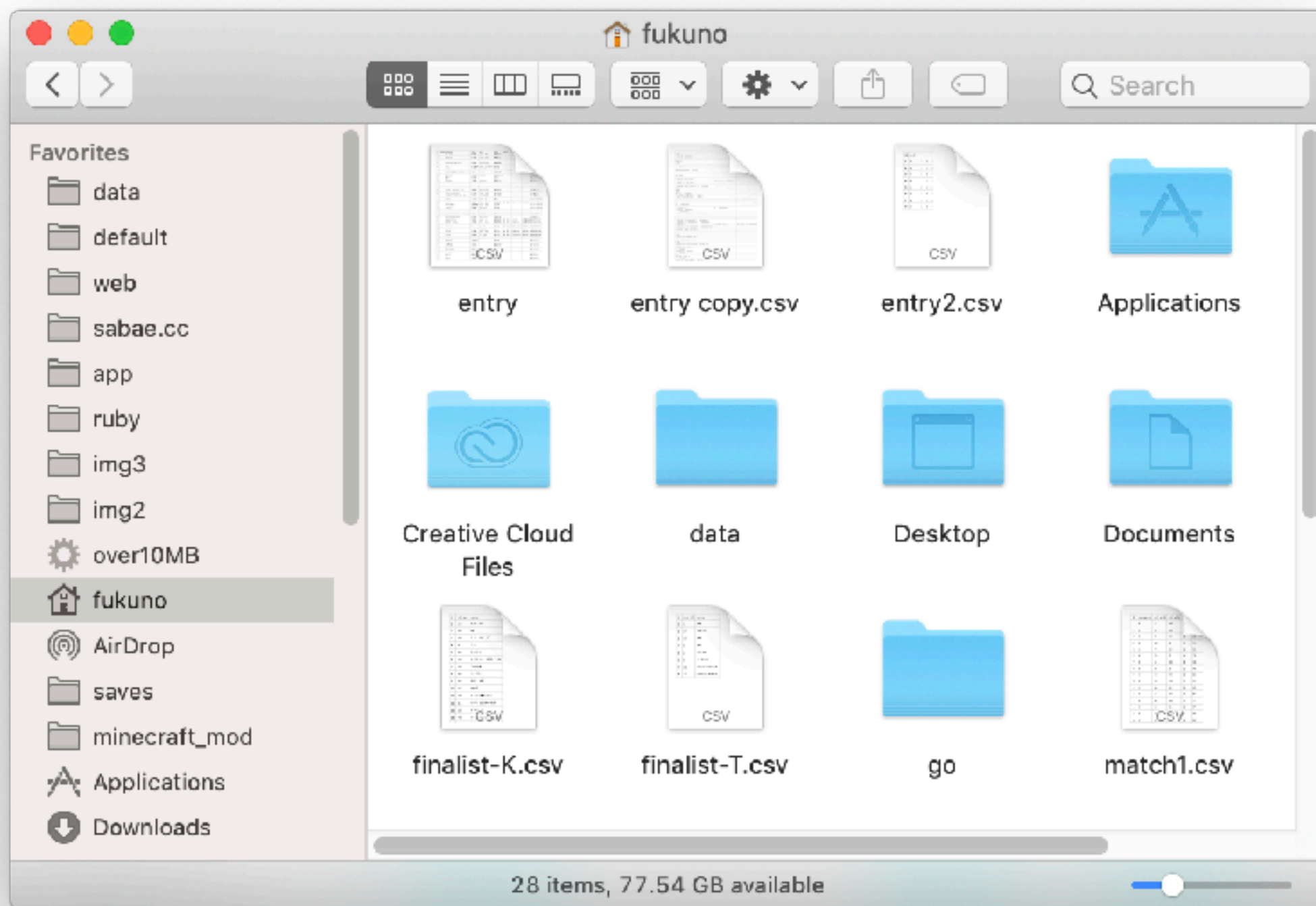
a



(イー、インター)

command not
found

(コマンド ノット ファウンド)



ホームフォルダのファイルを見せてもらおう

```
Last login: xxx
```

```
$a
```

```
-bash: a: command not found
```

```
$ls
```



L、S、インターキー（LIST）

change directory

「cd」フォルダをめぐるたび

```
cd ..
```

```
ls
```

```
cd /
```

```
ls -l
```

```
cd Users
```

```
ls -l
```

```
date
```

ひとつ上のフォルダへ

いちばん上のフォルダへ（ルート）

ハイフンエルで細かくみる

Users フォルダへ

（Uとかいて「タブキー」で楽！）

*おまけ、いまなんじ？

サーバーサイド開発にチャレンジ! by Node.js

<https://fukuno.jig.jp/2790>

福野泰介の一日一創

IchigoJam Sabae KidsIT opendata Maker KOSEN IoT megane profile english 検索

はじめてのウェブサービス、JavaScriptでつくる「チャット」チュートリアル Node.js編

2020/03/15 #js

webの本質は共同作業、みんなの力をキュートに集めるウェブサービス、自分でも創ってみたいですね。進化するプログラミングの世界、随分簡単にステキなサービスが作れるようになりました。

今回は、IchigoJam BASICからのステップアップにイチオシの言語「JavaScript」と便利ツール「Node.js」を使った「チャット」づくりのチュートリアル!



高専、春インターン!
エンジニア募集!
ふわっち
オタマート
IchigoJam
odp
Facebook
Twitter @taisukef
PV「感謝」
Code for Sabae
株式会社jig.jp



ホーム | NODE.JS とは | ダウンロード | ドキュメント | 参加する | セキュリティ | ニュース | 財団



Node.js® は、Chrome V8 の JavaScript エンジン で動作する JavaScript 環境です。

ダウンロード macOS (x64)

10.16.3 LTS

推奨版

12.12.0 最新版

最新の機能

他のバージョン | 変更履歴 | API ドキュメント | 他のバージョン | 変更履歴 | API ドキュメント

または、LTS のリリーススケジュールをご覧ください。

公式 Node.js ニュースレター Node.js Everywhere を購読しましょう。

 **LINUX FOUNDATION** COLLABORATIVE PROJECTS

Node.js の不具合を報告 | サイトの不具合を報告 | ヘルプ

© Node.js Foundation. All Rights Reserved. Portions of this site originally © Joyent.

Node.js is a trademark of Joyent, Inc. and is used with its permission. Please review the Trademark Guidelines of the Node.js Foundation.

Linux Foundation is a registered trademark of The Linux Foundation.

Linux is a registered trademark of Linus Torvalds.

Node.js Project Licensing Information.

<https://nodejs.org/>

Node.js をつかってみよう

```
$node
```

Node.js スタート！

```
> 3*5
```

「>」が node のしるし

```
15
```

計算してくれる！

```
> 2**16
```

いろんな計算！

```
> Math.PI
```

JavaScript が使える！

```
> Math.sqrt(2)
```

```
$
```

「Ctrl+D」で node おわり

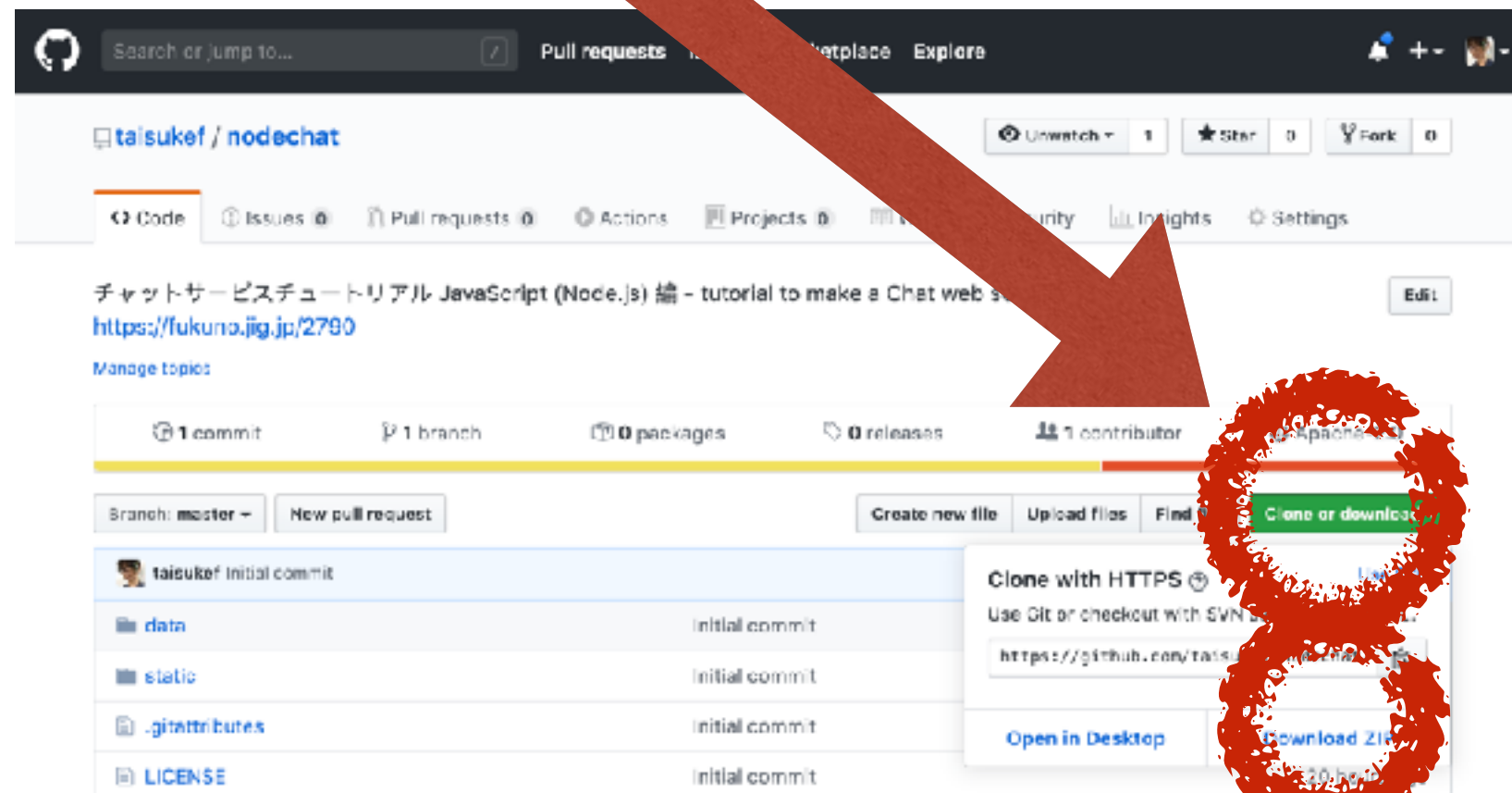
nodechatをダウンロード

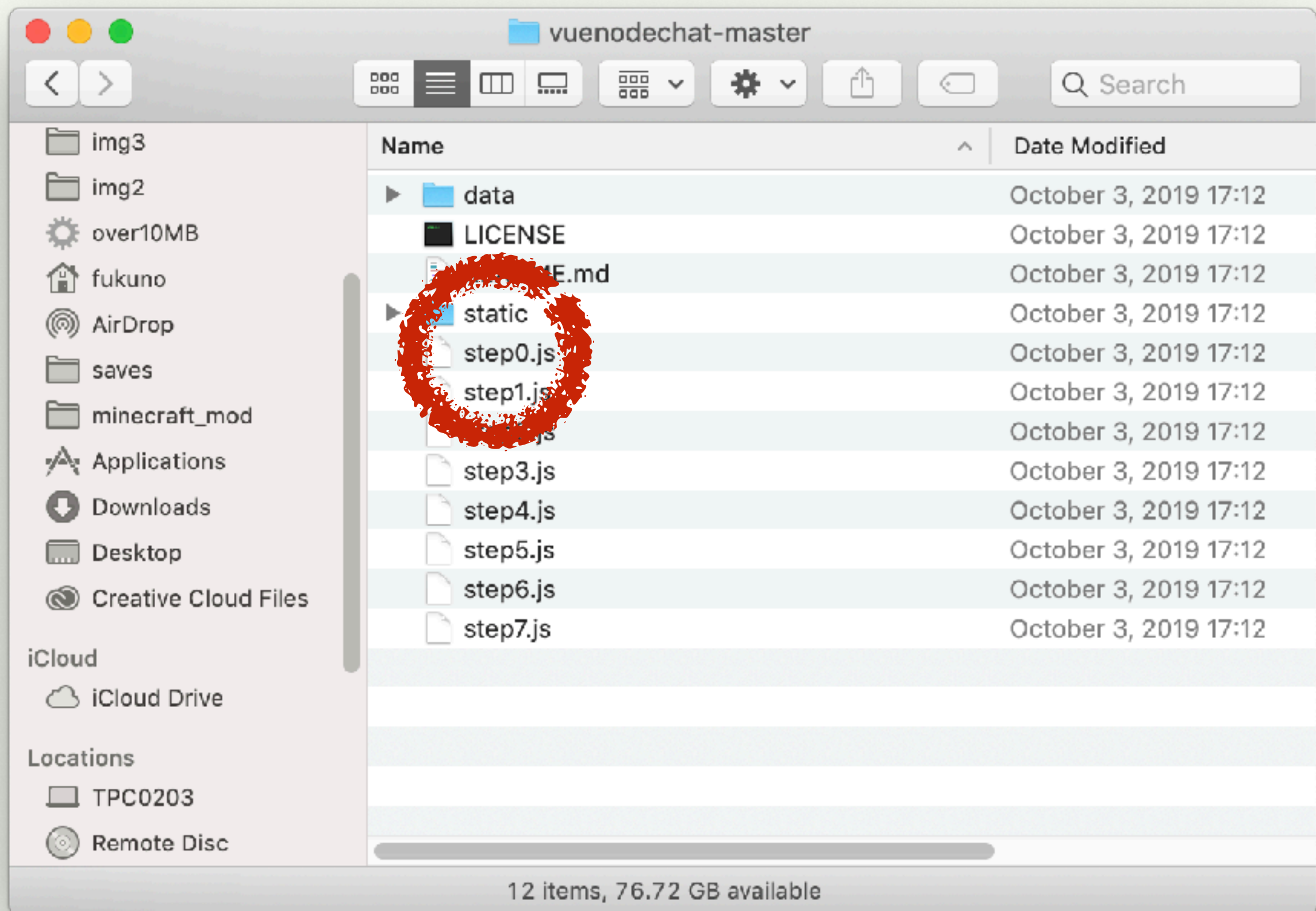
GitHubから今回のチャットサービス「nodechat」をダウンロード。
ダウンロードしたファイル、まずはJavaScriptで基本表示。console.logを使って、計算しましょう。(step0.js)

```
console.log(1 + 1)
```

<https://fukuno.jig.jp/2790>

ダウンロードできたら
zipファイルを展開する





「Atom」 や 「VSCode」 で 「step0.js」 を開こう

step0.js

JavaScriptのプログラムのなかみ

```
1 console.log(1 + 1)
```

```
2
```

`console.log(...)` は、中身を計算して表示するコマンド

IchigoJam BASIC の ? や PRINT

Python の `print(...)` に相当

ブラウザで動かす JavaScript でも使えるよ！

change directory

≡ ニ テ ク ニ ッ ク

```
cd |
```

1. cd、スペースまで打つ
2. いどうしたいフォルダをファイ
ンダーからドロップ
3. エンター

```
cd /Users/fukuno/Downloads/nodechat-master |
```

Node.js をつかってみよう

```
$node step0.js
```

```
2
```

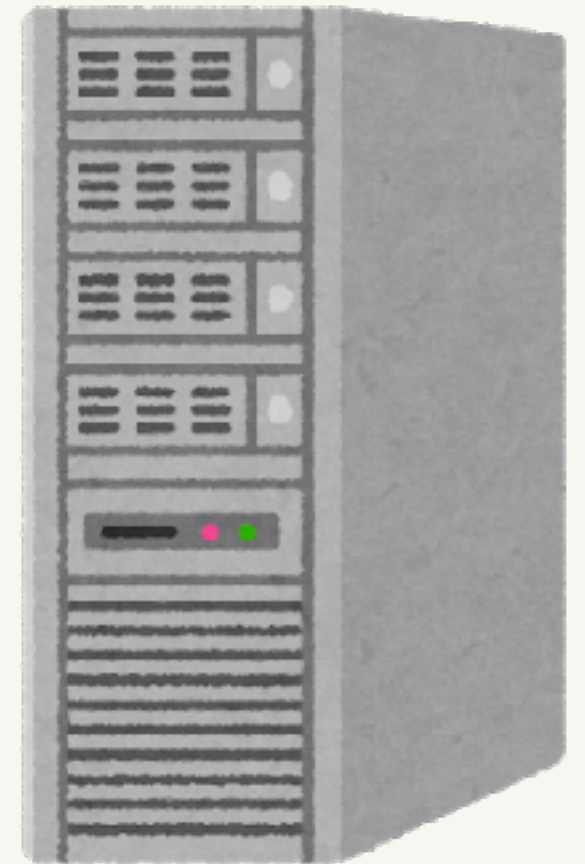
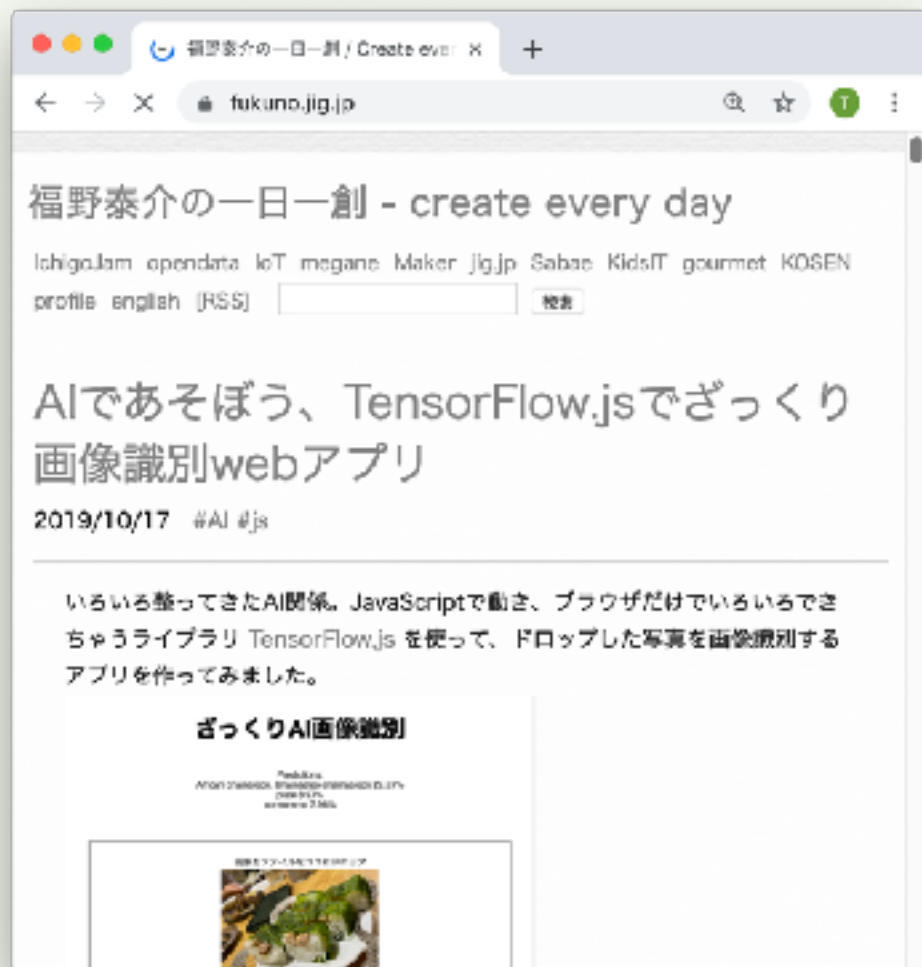
```
$
```

JavaScript でつくったプログラ「step0.js」
でNode.js を起動！

step0.js の console.log のなかをカイゾウし、
3*5 を計算してみよう

Webのしくみ（ブラウザとサーバー）

ブラウザはURL内のホストで指定されたコンピューターへ
送って欲しいファイル名（パス）をリクエスト



サーバーはHTMLや画像などをレスポンス

Webサーバーをつくろう

続いて、ウェブサーバーを作ります。(step1.js)

<https://fukuno.jig.jp/2790>

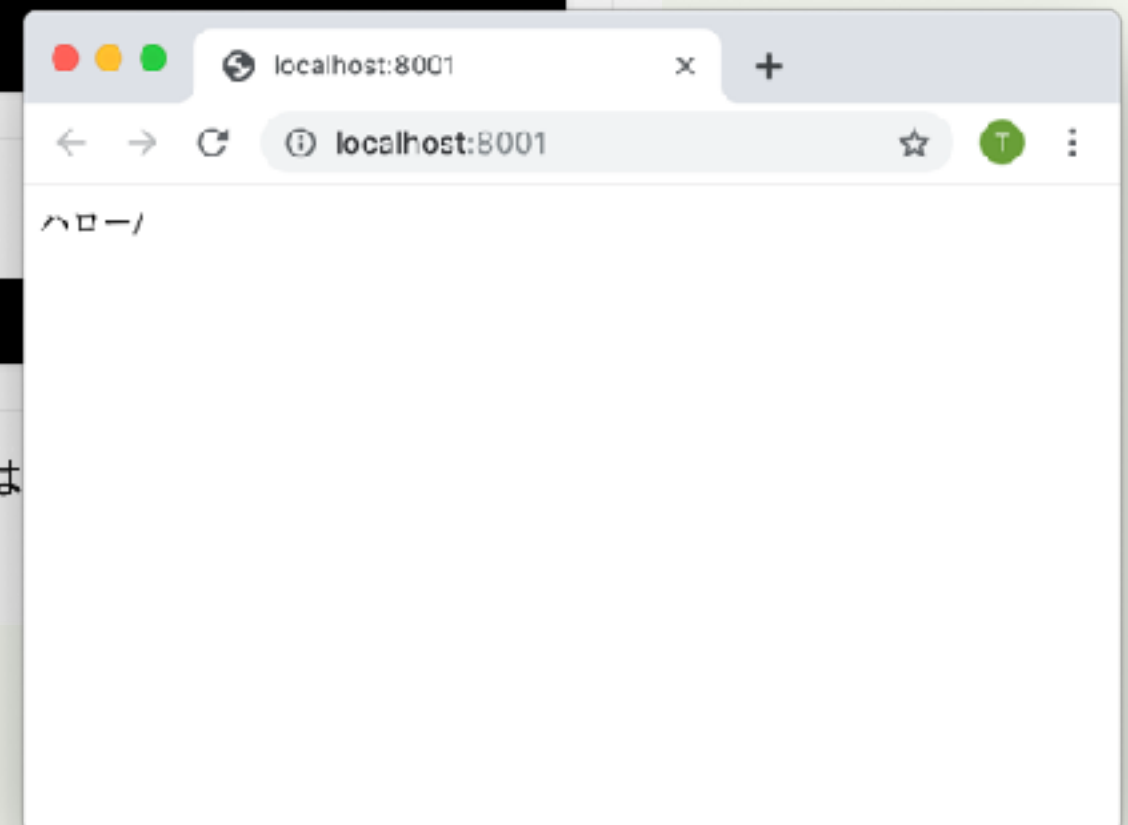
```
const http = require('http')

const server = http.createServer()
server.on('request', function(req, res) {
  console.log(req.url)
  res.writeHead(200, { 'Content-Type': 'text/html; charset=utf-8' })
  res.write('ハロー' + req.url)
  res.end()
})
server.listen(8001)
```

短い！こちらも同じように動かします

node step1.js

も出ずに固まっている？エラーが出てなければ無事動いているは
<http://localhost:8001/>を開き



<http://localhost:8001/> を開く

webサーバーをカイゾウしよう

ATOM/VSCodeで step1.js を編集

```
$node step1.js
```

nodeをとめるときは **Ctrl+C**

カーソル「上」押して、エンターでもういちど！

step0.js の console.log のなかをカイゾウし、
3*5 を計算してみよう

その調子！

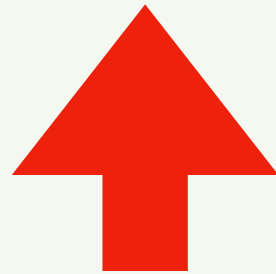
step4 まで”すすめてみよう

<https://fukuno.jig.jp/2790>



ともだちとつないで"あそぼう"！

```
$ifconfig | grep br0  
inet 172.xxx.xxx.xx netmask
```

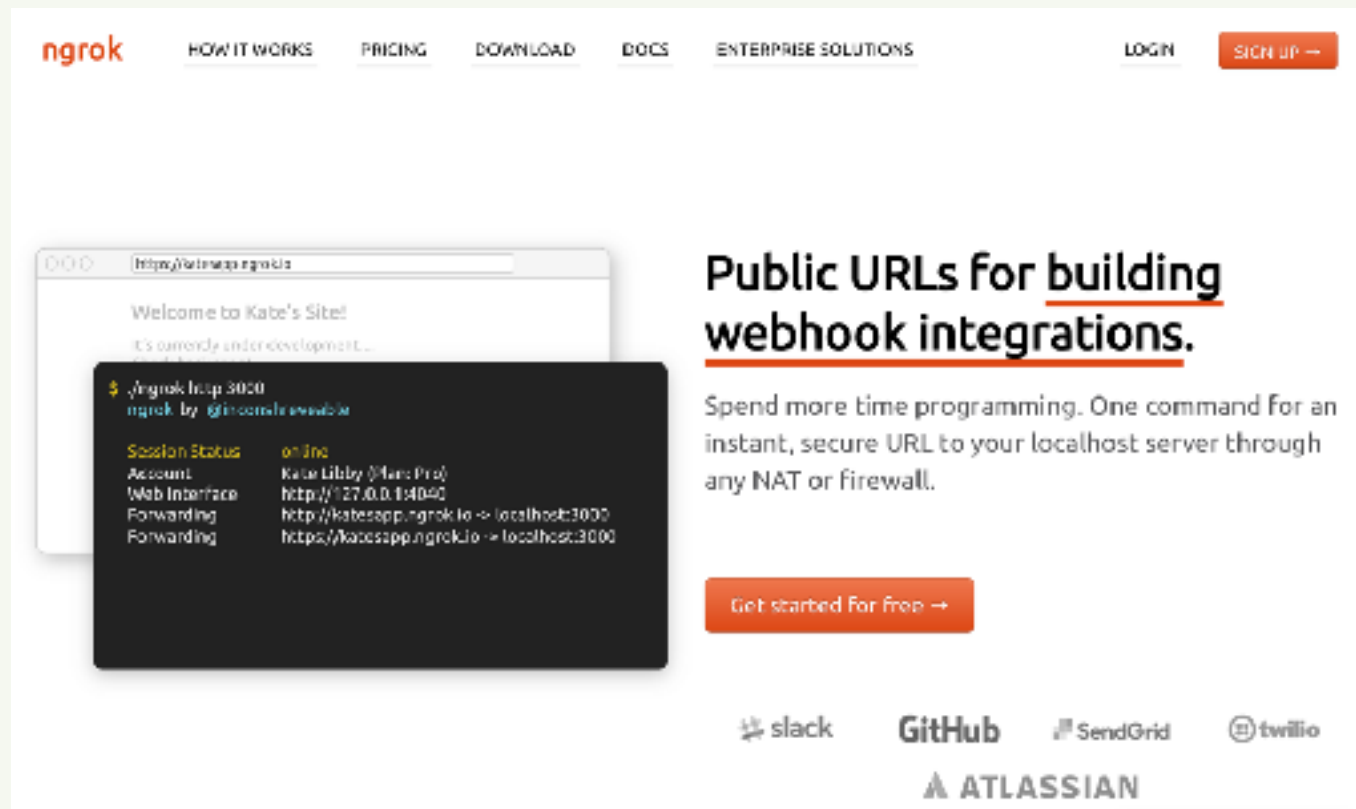


自分のコンピューターのIPアドレス

[http://\[IPアドレス\]:8001/chat.html](http://[IPアドレス]:8001/chat.html)

このURLを友だちに教えてつないでみてもらおう
(同じLANにつながったスマホでもOK！)

ngrok（エングロック）で LAN外のスマホともつながろう



<https://ngrok.com>

「ngrok」をダウンロード

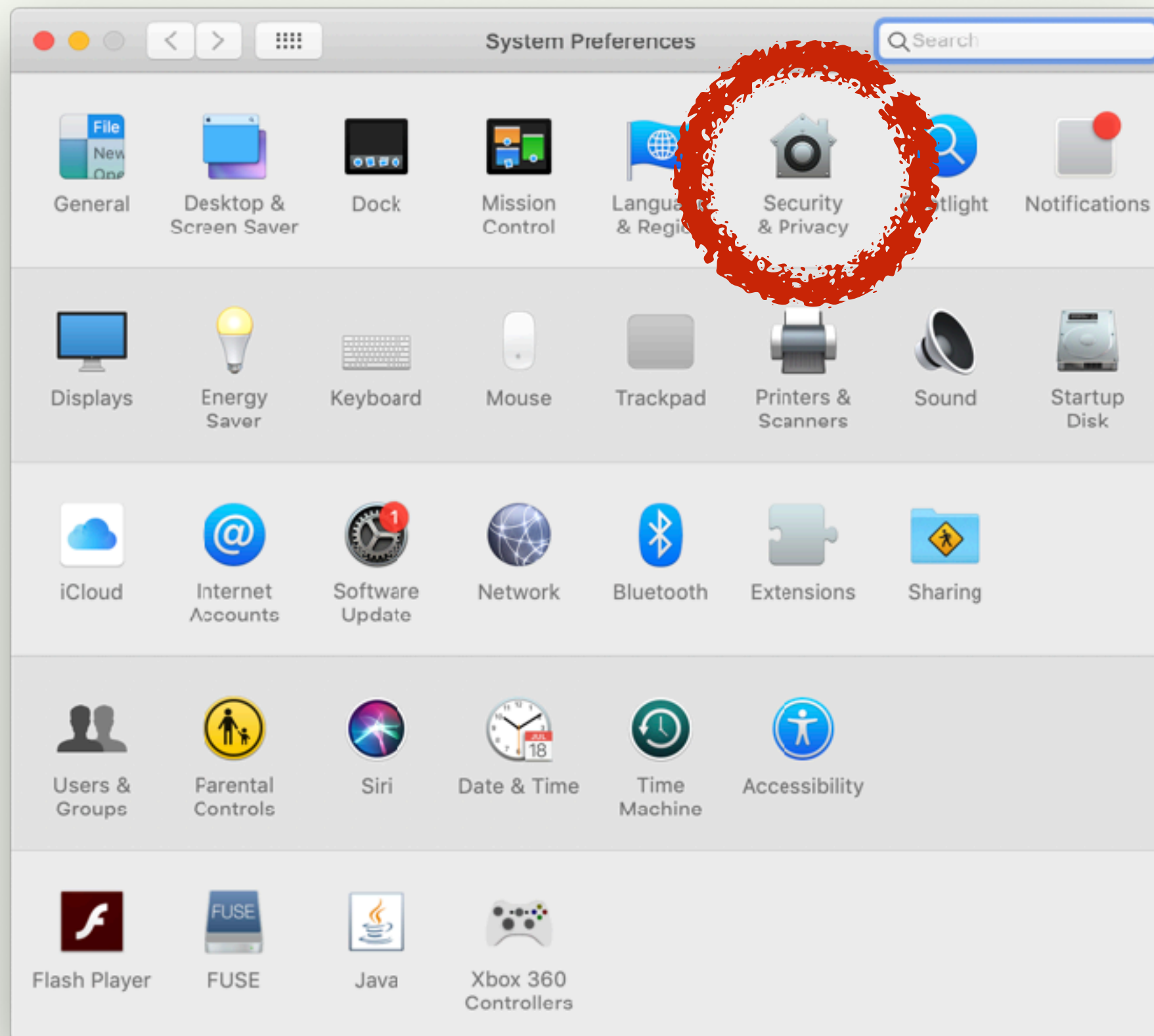
ターミナルをもうひとつ開き

```
$sudo mv ngrok /usr/bin/
```

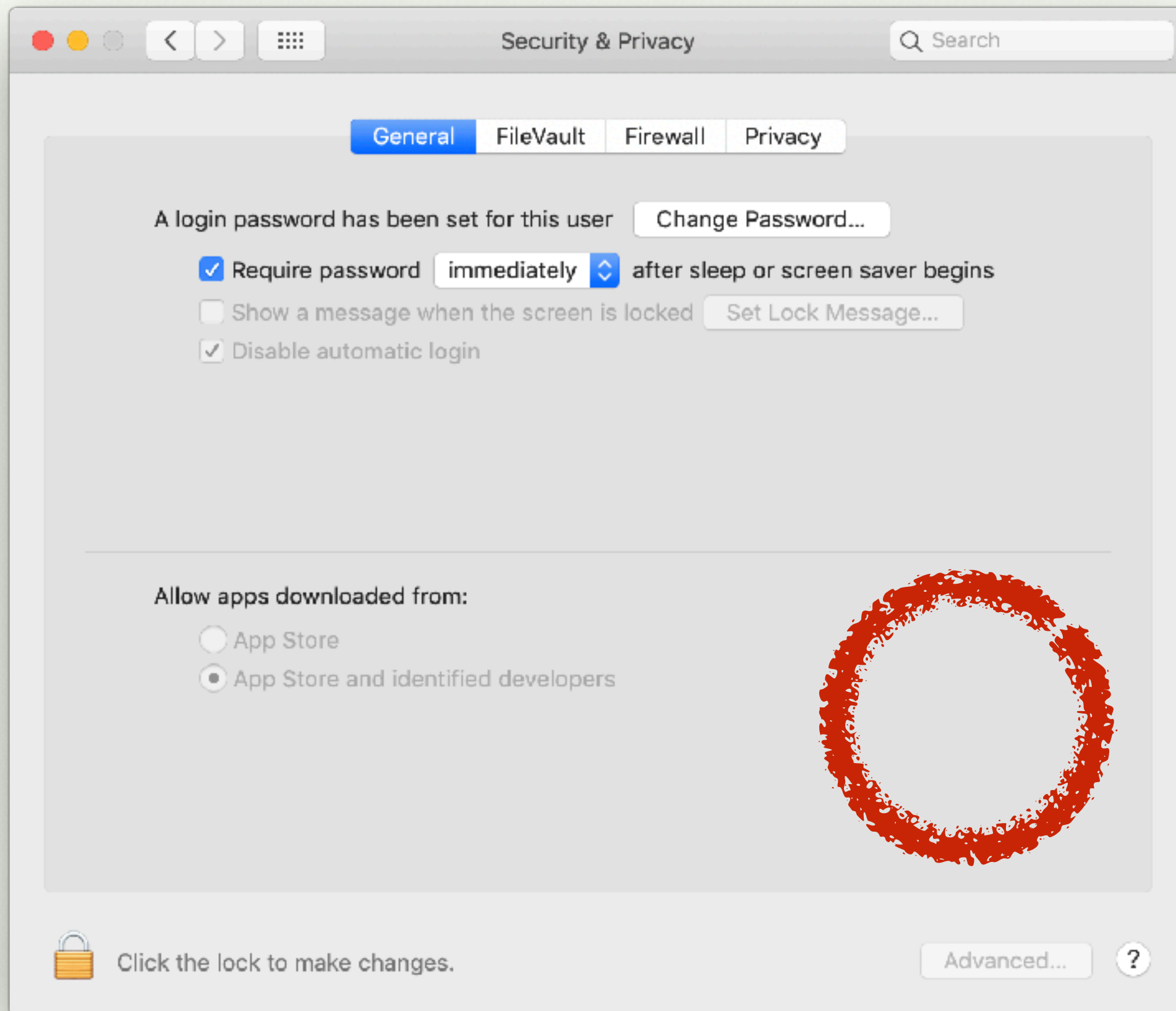
Macのログインパスワード入れる

```
$ngrok http 8001
```

警告がでたらセキュリティ設定
詳細は次のスライド



もしセキュリティの関係で開けなかったら



ここで「開く」を選択する