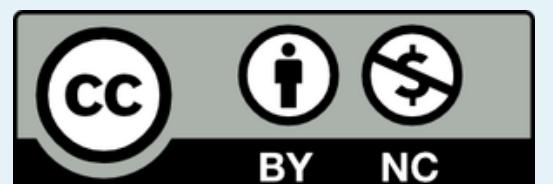


# メディアアート・プログラミング2

東京藝術大学 芸術情報センター開設科目 後期金曜4限 第1週

2023.10.06 松浦知也 ([matsura.tomoya@noc.geidai.ac.jp](mailto:matsura.tomoya@noc.geidai.ac.jp) [teach@matsuuratomo.ya.com](mailto:teach@matsuuratomo.ya.com))



# 再掲：Google Classroom

ujriogo

履修人数的に抽選は無し、  
CampusPlanでの登録を忘れずに



CUIと10 PRINT

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

# 10 PRINT



<https://10print.org/>

オンラインPDFはCreative Commonsライセンスで無料で読める

もしデスクトップ環境なしで、  
マウスもなしでコンピューターを操作  
することになったら？

# Commodore 64

- コモドール社が1982年1月に発表した8ビットホームコンピューター
- 「ホーム」コンピューターではあるけど、今日のようなデスクトップ環境が登場するよりは前
- 音楽では、チップチューンに使われたコンピューターとしても有名
- BASICという言語でプログラミング可能
  - サンプルとして10 PRINT~が含まれていた



**C64 online emulator** for bringing back your childhood memories or just for enjoying 8-bit retro classics. Thanks to this online emulator you can have a Commodore 64 right inside your browser without installing any software. You can drag & drop your files inside the box under C64. Accepted formats include D64, G64, X64, S64, ZIP, PRG, P00, T64, TAP and also CRT. Joystick simulation and USB gamepads are also supported. If you don't have C64 games to play, visit our [Commodore 64 game library](#) here.



<https://c64online.com/c64-online-emulator/>

# エミュレーターで実行してみよう

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

コピペとかできない&キーボードのレイアウトが違うかもしれない注意

以下はANSI USキーボードの場合

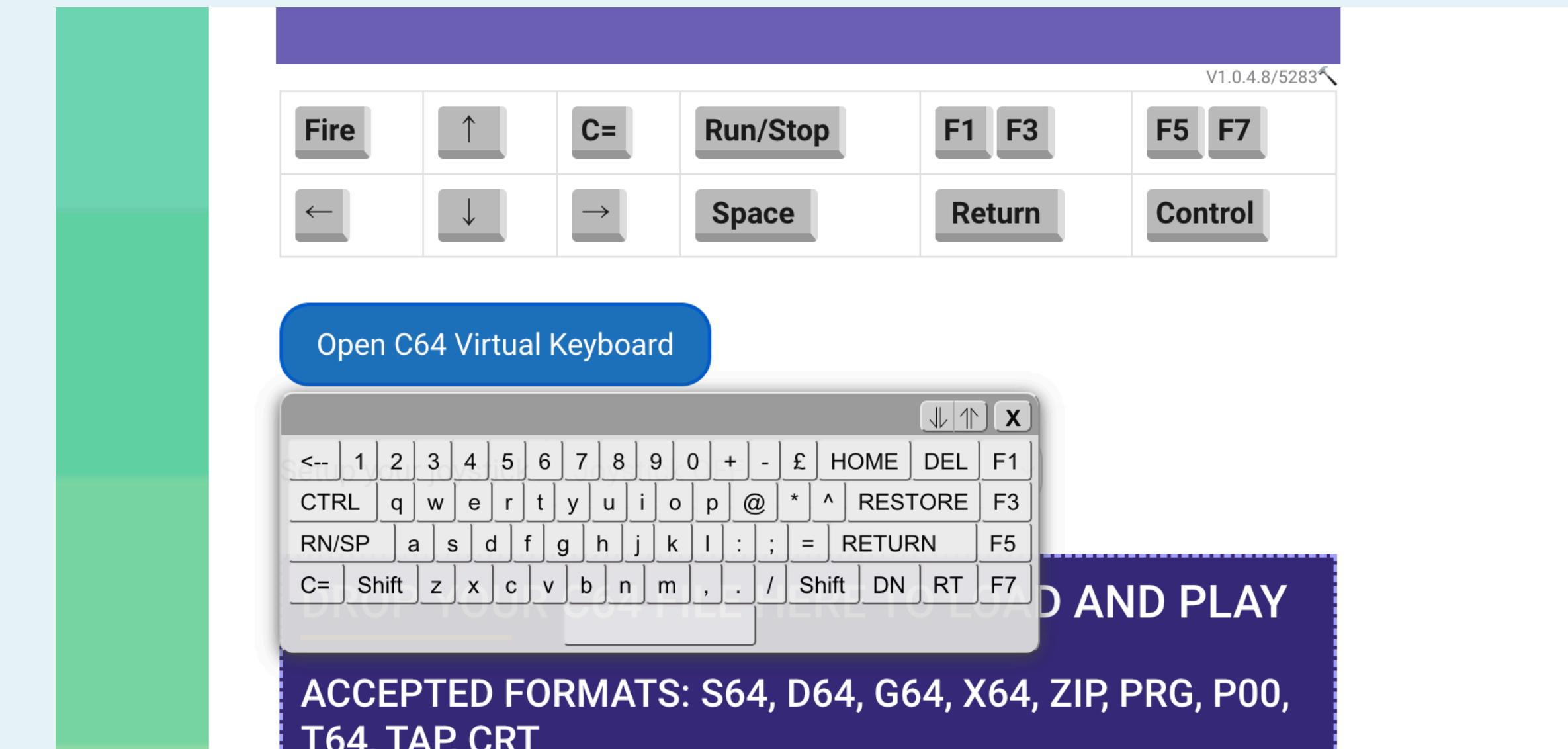
\$ → : shift+4

() → shift+ 8/9 (9/0じゃない！)

; → ;の1つ右隣

: → ` (1の左隣)

JISキーボードの場合、セミコロンが打てないのでブラウザ上のバーチャルキーボードを押して入力しよう



# エミュレーターで実行してみよう

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

入力できたら改行して

RUN

を実行

game history here.

Fullscreen

Sound

Save

Load



V1.0.4.8/5283 ↵

Fire	↑	C=	Run/Stop	F1 F3	F5 F7
←	↓	→	Space	Return	Control

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

↑10はプログラムの行番号

GOTO {行番号}で特定の行へ処理をジャンプ↑

全体としては無限ループを作っている

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

↑10はプログラムの行番号

GOTO {行番号}で特定の行へ処理をジャンプ↑

大きな処理の単位ごとに10,20,30...とか番号を振っていって、途中で  
処理を増やしたくなったら12,13など細かい番号をつけていく慣習

全体としては無限ループを作っている

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

↑10はプログラムの行番号

GOTO {行番号}で特定の行へ処理をジャンプ↑

大きな処理の単位ごとに10,20,30...とか番号を振っていって、途中で  
処理を増やしたくなったら12,13など細かい番号をつけていく慣習

全体としては無限ループを作っている

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

↑PRINTは続く式を標準出力に書き込む

↑CHR\$()は文字コードの数値から文字を作る関数

↑RND()は0から指定した値までの乱数を作る関数

```
10 PRINT CHR$(205.5+RND(1)); : GOTO 10
```

↑の計算結果はCHR\$に読まれる時に整数化され、  
50%の確率で205か206のどちらかに  
Commodoreでは205が＼、206が＼という文字に対応する

# ターミナルでの操作入門

# ターミナルで使えるコマンド

- おおよそmacOSとLinux系の様々なOSでは共通したコマンドが使える
- これはmacOSとLinuxがUNIXという共通の祖先を持つため（詳しい歴史的経緯は複雑なので割愛）

ITmedia NEWS > 企業・業界動向 > 「MacはLinuxだった」って本当？「そうだね……」…

ヤマーとマツの、ねえこれ知ってる？

## 「MacはLinuxだった」って本当？「そうだね……」元Mac雑誌編集者は昔語りを始めた

2022年01月26日 12時00分 公開 [松尾公也, 山川晶之, ITmedia]

印刷 通知 見る Share B! 766 15

PR あなたの職務適性が15分でわかる | AMBI (アンビ)  
PR 「360°マイク」はオンライン会議でどんなふうに便利なのか？

経歴だけは長いベテラン記者・編集者の松尾（マツ）と、幾つものテック系編集部を渡り歩いてきた山川（ヤマー）が、ネット用語、テクノロジー用語で知らないことをお互い聞きあったり調べたりしながら成長していくコーナー。交代で執筆します。

マツ 「MacはLinuxだった」という趣旨のツイートが話題になっていて、ああそういうこともあったなあ、と思い出したので、ちょっと昔語りしようかなと。茶でも飲みながら。

ヤマー 私は紅茶を用意してきます。

<https://www.itmedia.co.jp/news/articles/2201/26/news077.html>

# 基本的なお約束

- ファイル名を除いて半角英数で入力が基本
- 大文字小文字は区別される
- Enterを押すとコマンド実行
  - 複数行にわたるコマンドの使い方はそのうち説明
- 入力前に↑↓を押すことで実行したコマンドの履歴を辿れる
- プログラムを実行中に止めたくなった場合は^(control)+c
  - Node.jsやPythonなど一部対話環境は^(control)+d

# 基本的なコマンド

ls	List	ファイル一覧の表示	ls hogeでhogeフォルダの中を表示
cd	Change Directory	ディレクトリの移動	./で現在のフォルダを探す ../でひとつ上のフォルダを探す
mv	Move	ファイルの移動 名前変更	ファイルの名前変更＝ 同じフォルダの違う名前へ移動
touch		ファイルの作成	
mkdir	Make Directory	フォルダの作成	
rm	Remove	ファイル・フォルダの削除	rm -rf hoge でhogeフォルダの中身を全て削除

# 基本的なコマンド（続き）

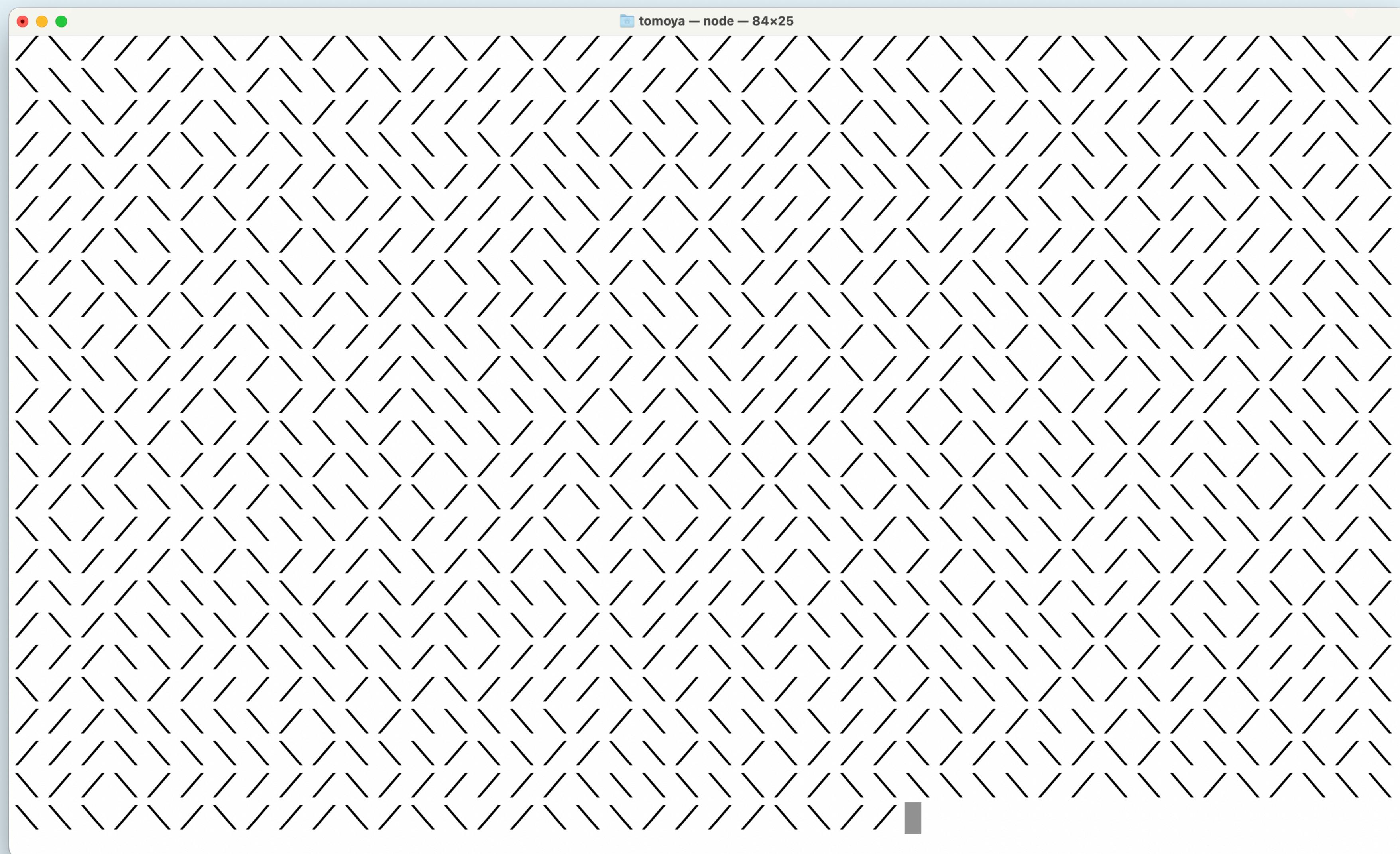
open		システムでファイルを開く	Finderでダブルクリックしたのと同じ効果
cat		ファイルを標準出力に表示	大きいファイルはheadやtailコマンドで最初と最後だけを見ることも
echo		標準出力に文字列を表示	他のコマンドとの組み合わせで使う

# JavaScriptで10 PRINT

# Node.jsのインストール

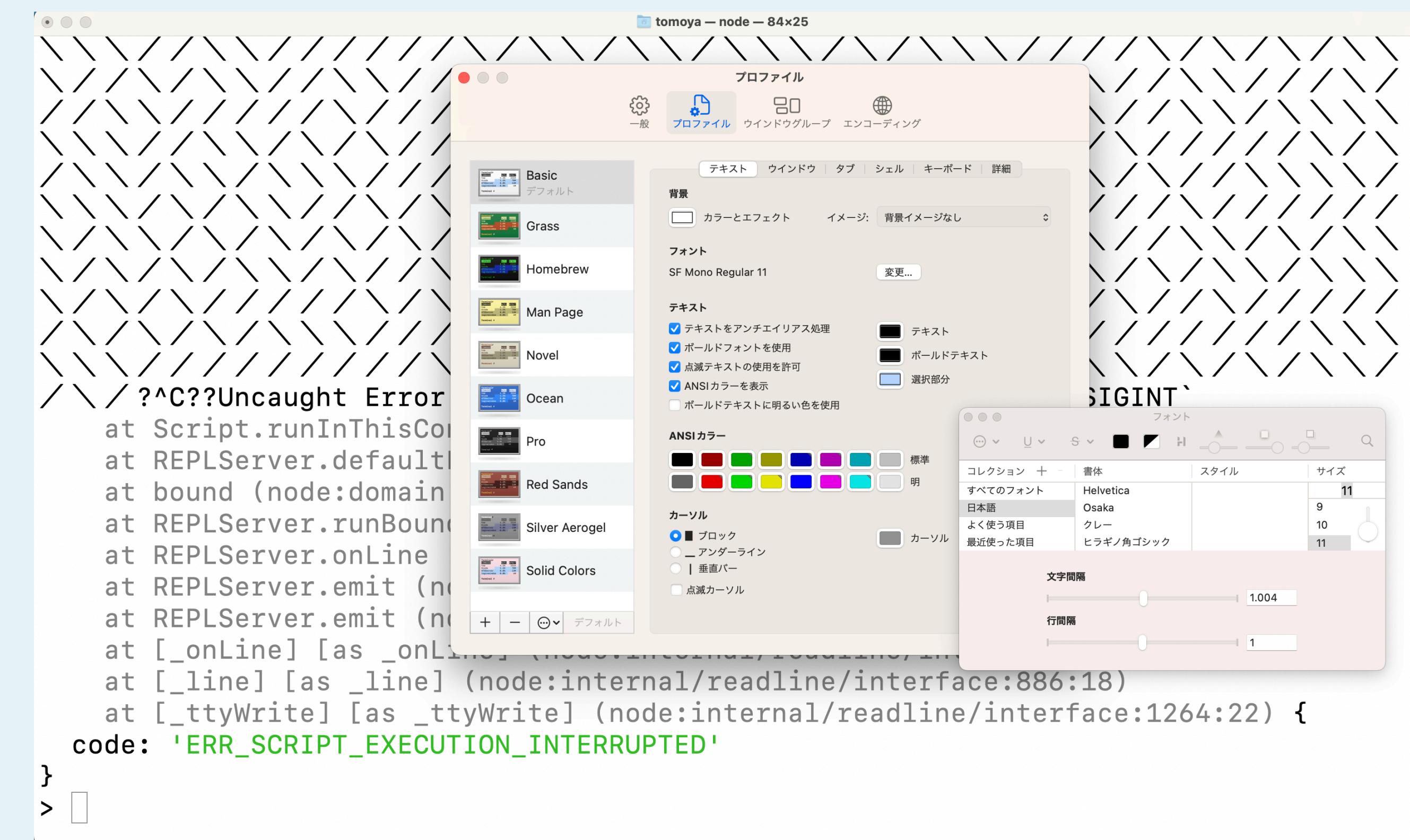
```
brew install nodejs
```

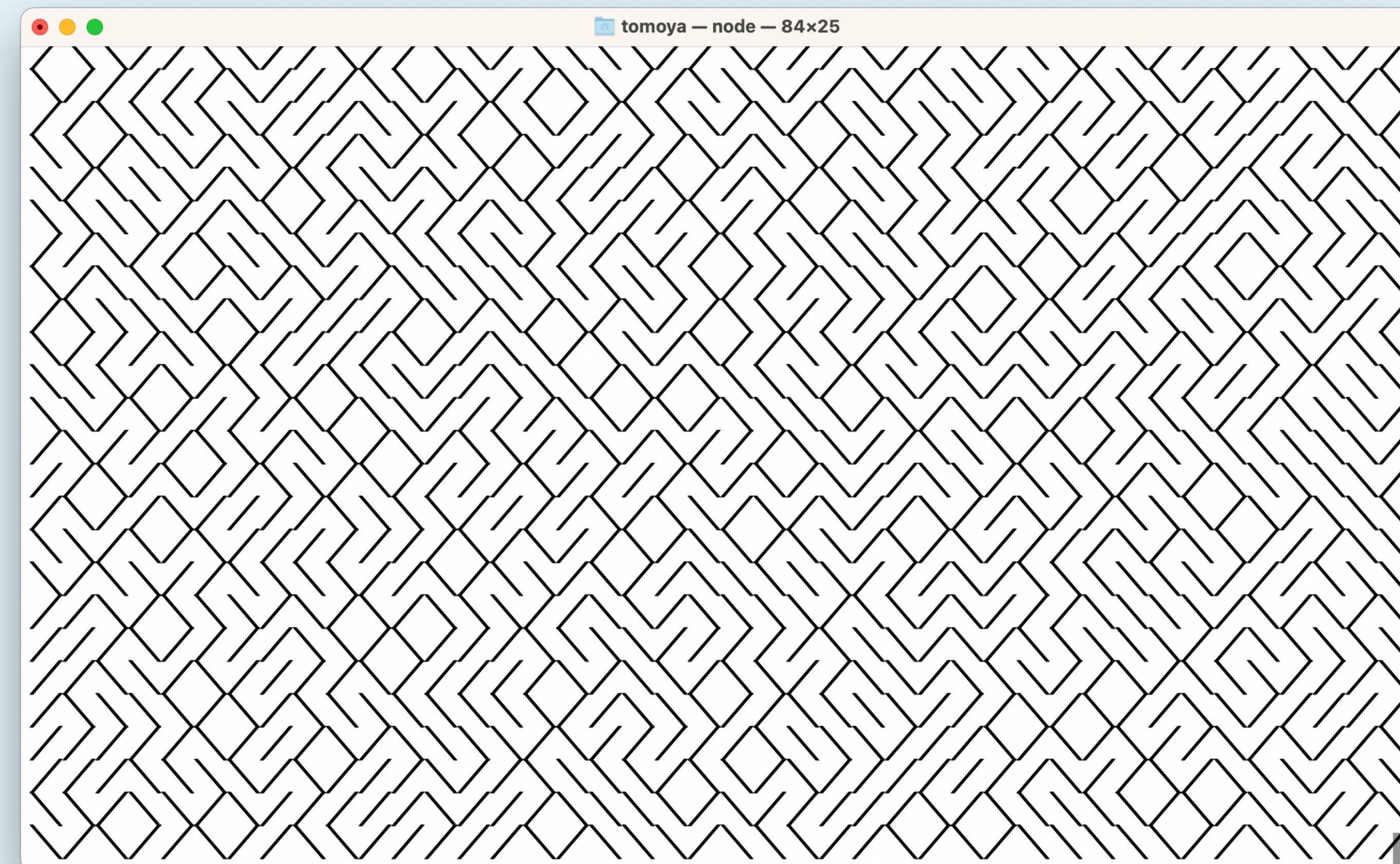
- ・ インストールされるとnodeコマンドでNode.jsの対話環境が使えるようになる
- ・ Node.jsはWebブラウザ上で動くプログラミング言語のJavaScriptをローカルの環境で使えるようにしたもの



# ターミナルの見栄え調整

- Cmd + / Cmd -で文字の大きさ調整
- 環境設定→プロファイル→フォント 変更...→文字間隔・行間隔をそれぞれ0.7ぐらいまで縮めてみよう





# Node.jsでの10 PRINT(丁寧版)

```
while(true){  
    if(Math.random(1)>0.5){  
        process.stdout.write("\\" )  
    }else{  
        process.stdout.write("//")  
    }  
}  
  
// while(true) process.stdout.write( Math.random(1)>0.5? "\\" : "//" )
```

# Node.jsでの10 PRINT(丁寧版)

```
while(true){  
    if(Math.random()>0.5){  
        process.stdout.write("\\" )  
    }else{  
        process.stdout.write("//")  
    }  
}  
  
// while(true) process.stdout.write( Math.random()>0.5? "\\" : "//" )
```

このテキストファイルを、10print.jsとして保存して  
node 10print.jsとして実行できる

```
setInterval(  
  ()=>{  
    process.stdout.write( Math.random(1)>0.5? "\\" : "/" )  
}, 50)
```

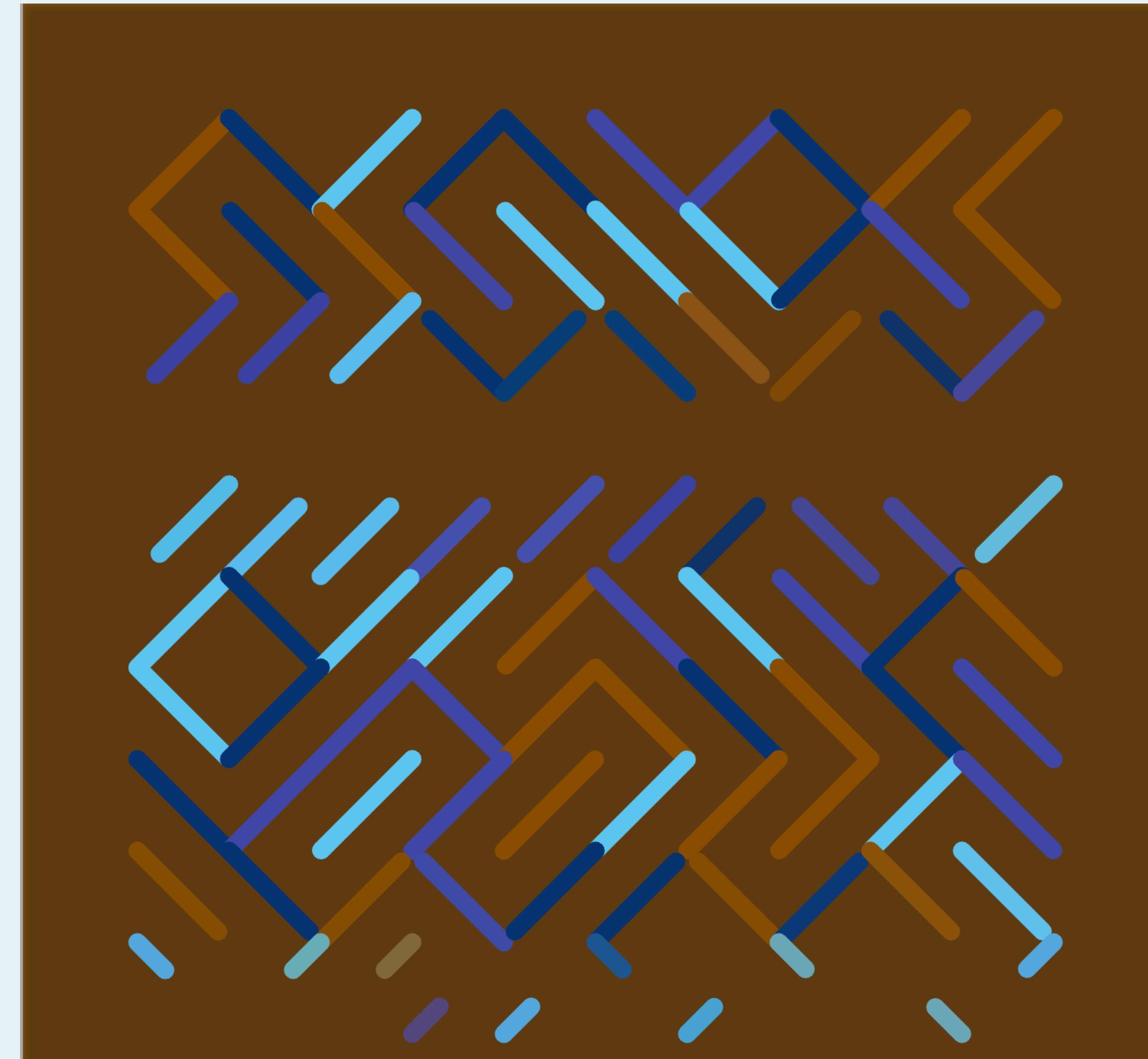
while(true)による無限ループの代わりに定期実行の関数setIntervalを使って速度を調整

# 応用例

- 確率を50%：50%から変えてみよう
- スラッシュ以外の別の文字を使ってみよう
- Processingなどグラフィック環境で似たようなことをやってみよう
  - 線の太さや長さにもランダム性を与えてみたら？
  - 線の代わりに画像でタイルしたら？

# Processingでの応用例

takawo shunsuke



<https://openprocessing.org/sketch/2029378>

# 補足：codeコマンドのインストール

- Visual Studio Codeで Cmd+Shift+Pでコマンドパレットを開き、"code"と検索
- PATH内に'code'コマンドをインストールしますを選択
- openと似たような形でcode {ファイル名}でファイルを VSCodeで開けるようになる

