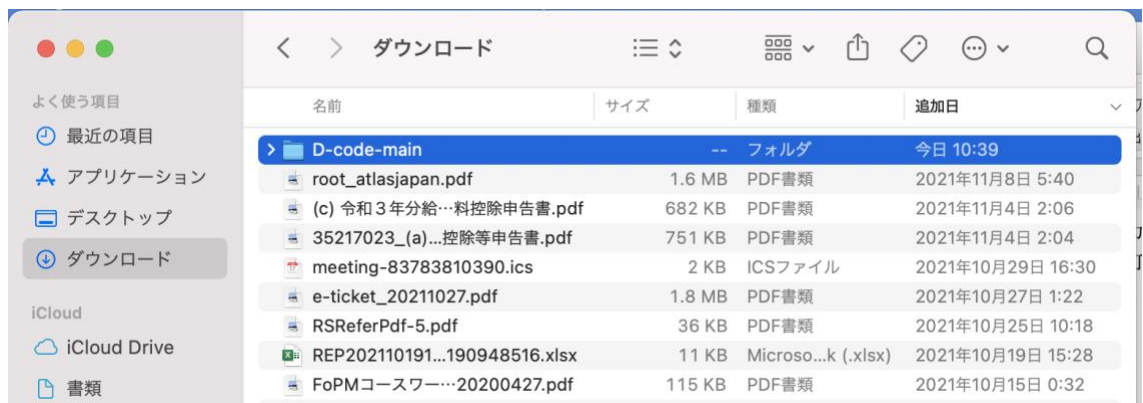


# Disney Ticket Getter

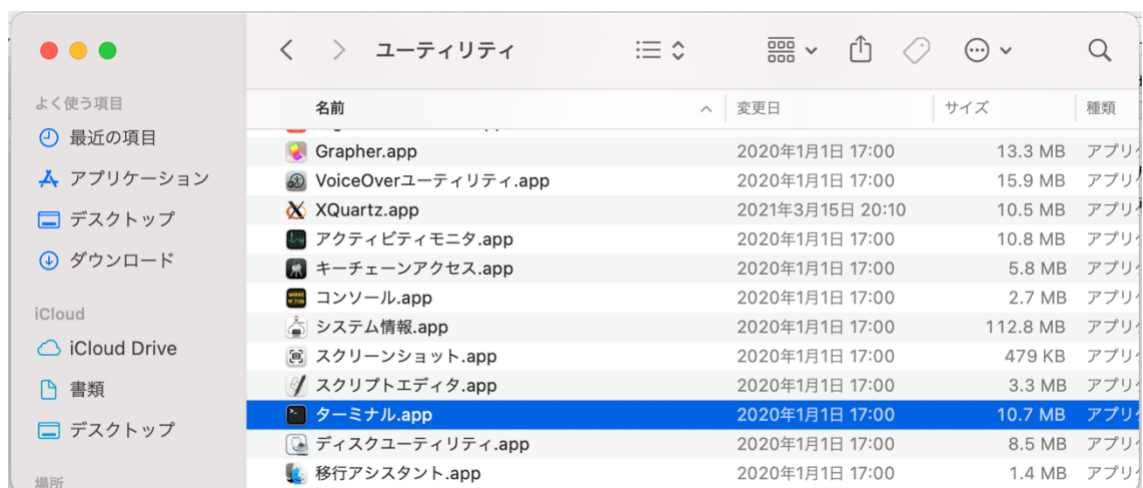
## コードマニュアル(Mac 版)

### ・使い方

1. パソコンで GitHub を開き、コードをダウンロードしてください。Finder のダウンロードに”D-code-main”が追加されるので好きなディレクトリに移してください（別に移さなくても問題はないですが）。



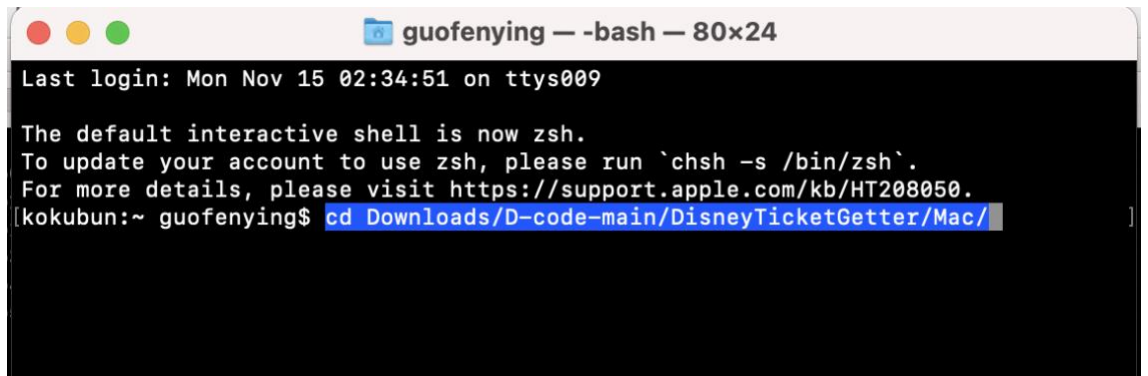
2. Finderで「アプリケーション」→「ユーティリティ」と移動し、「ターミナル.app」を起動します。



3. 開いたターミナル上で `cd` コマンドを使ってコードが置いてあるディレクトリに移動します。1. で特に移動しなかった場合は下の写真と同様に

`" cd Downloads/D-code-main/DisneyTicketGetter/Mac/"`

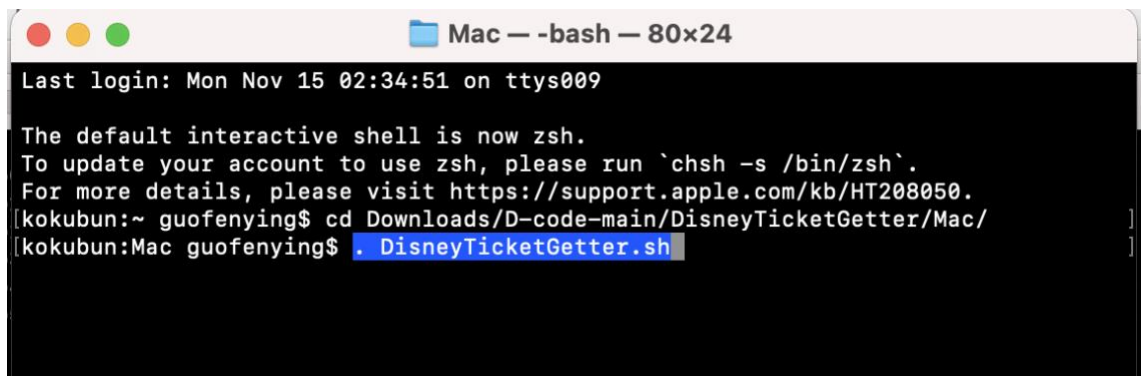
と打てばいけるはずです。



```
guofenying — -bash — 80x24
Last login: Mon Nov 15 02:34:51 on ttys009

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[kokubun:~ guofenying$ cd Downloads/D-code-main/DisneyTicketGetter/Mac/
```

4. “. DisneyTicketGetter.sh”と打てば実行できます。（先頭に.と半角スペースがあるのに注意してください）



```
Mac — -bash — 80x24
Last login: Mon Nov 15 02:34:51 on ttys009

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[kokubun:~ guofenying$ cd Downloads/D-code-main/DisneyTicketGetter/Mac/
[kokubun:Mac guofenying$ . DisneyTicketGetter.sh
```

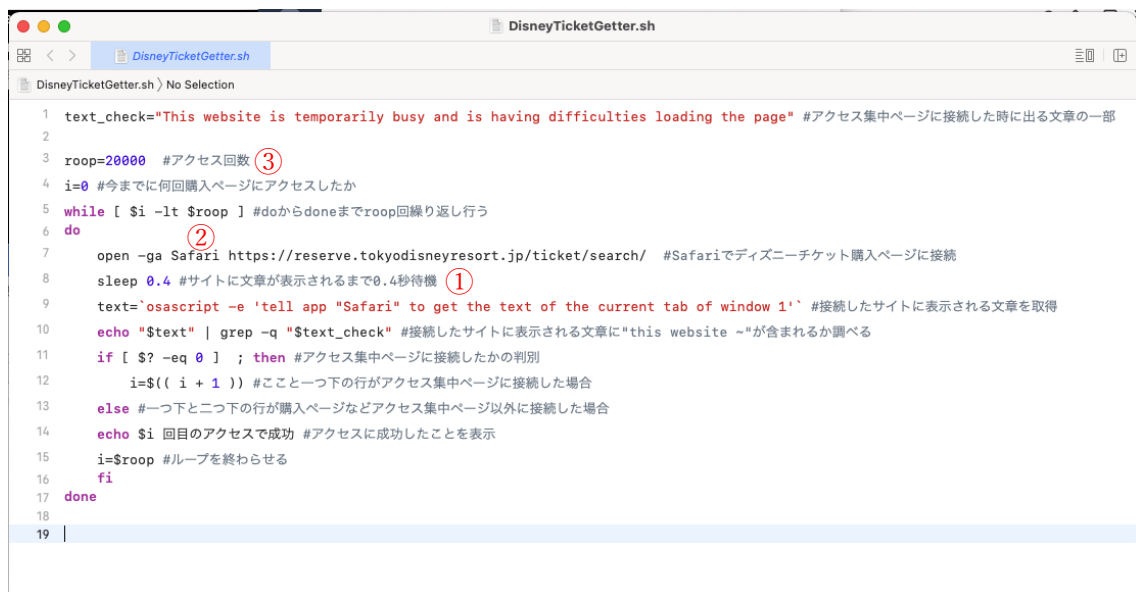
5. 途中で止めたい時は“Ctrl+c”を入力してください。（このコード君はデフォルトだと約2時間ディズニーのサイトにアタックし続けます）

## ・ 注意点

- ・ コードが走っている最中に Safari で違うタブに移動したり、新しいウィンドウを開いたりするとコードが止まります。Chrome や Firefox 等 Safari 以外のウェブブラウザを使うか、後述する設定変更で Safari 以外のブラウザでコードを走らせるようにしてください。
- ・ アクセス頻度はデフォルトで 2.5 回/秒(0.4 秒に 1 回)です。僕のパソコンでは問題ないのですがもしかしたらディズニーのサイトから Access Denied されるかも知れません。その場合、設定変更してアクセス頻度を下げてください。

## ・ コードの設定変更をしたい場合

設定を変更したい場合、Finder で”DisneyTicketGetter.sh”をダブルクリックしてテキストエディタを使うか、ターミナルで emacs や vim を使っていじることになります。



```
1 text_check="This website is temporarily busy and is having difficulties loading the page" #アクセス集中ページに接続した時に出る文章の一部
2
3 roop=20000 #アクセス回数 ③
4 i=0 #今までに何回購入ページにアクセスしたか
5 while [ $i -lt $roop ] #doからdoneまでroop回繰り返す行
6 do
7     open -ga Safari https://reserve.tokyodisneyresort.jp/ticket/search/ #Safariでディズニーチケット購入ページに接続
8     sleep 0.4 #サイトに文章が表示されるまで0.4秒待機 ①
9     text=`osascript -e 'tell app "Safari" to get the text of the current tab of window 1` #接続したサイトに表示される文章を取得
10    echo "$text" | grep -q "$text_check" #接続したサイトに表示される文章に"this website ~"が含まれるか調べる
11    if [ $? -eq 0 ] ; then #アクセス集中ページに接続したかの判別
12        i=$(( i + 1 )) #ここ一つ下の行がアクセス集中ページに接続した場合
13    else #一つ下と二つ下の行が購入ページなどアクセス集中ページ以外に接続した場合
14        echo $i 回目のアクセスで成功 #アクセスに成功したことを表示
15        i=$roop #ループを終わらせる
16    fi
17 done
18
19 |
```

開くと上の写真のようなコードが出てきます。簡単なので少しいじれば変更できます。各行の説明はコメントアウトに書いてありますが、必要になりそうな変更の仕方について書いておきます。

- ① (3 行目)アクセスに何回失敗したらあきらめるかをここで決めています。デフォルトは 2

万回で、2 時間弱がんばってくれます。もっと頑張れやっ場合は増やしてください。

- ② (7 行目)Safari でディズニーチケット購入ページを開いてくれようとしてくれます。Safari 以外のウェブブラウザでコードを走らせたい場合には 7 行目と 9 行目にあ  
る”Safari”の文字を”Chrome”やら”Firefox”やらに変更してください。ディズニー以外  
のサイトに使いたい場合には 1 行目のテキストと 7 行目の URL を変更すれば使えま  
す。
- ③ ここでサイトに文章が表示されるのを待つと同時に、アクセス頻度を決定していま  
す。短くすればするほどコードくんの効率は高まりますが、ある一定頻度を超えると  
Access Denied されるので、ちょうどいいところを決めなければなりません。(僕のパソ  
コンでは 0.4 では問題なく、0.35 だといけたりいけなかったり...あんまり詳しく検証  
してないのでよりタイムが見つかったら教えてください)

その他、したい変更や欲しい機能などあれば言ってください (14:00 になったら自動で起  
動する機能とか、アクセス成功したら音で知らせてくれる機能とか?)。やれる範囲で対  
処します。