

# Ruby おためし

Ruby 語で書かれたプログラムを動かしてみましょう.

その前に Ruby の準備とサンプルの準備が必要かも

・ご自分の PC が Windows の人は **Ruby お手軽準備** からどうぞ

Mac 人 (= PC がアップル社の Mac の人) へ

<http://tcs.c.titech.ac.jp/cs/jyunbi.pdf>

おめでとうございます！Ruby の準備は不要です

- ・サンプル(プログラムと文)は <http://tcs.c.titech.ac.jp/cs/sample.zip> から zip ファイルをデスクトップ上にダウンロードして解凍しておいてください
- ・まあ、ダウンロードが面倒な人は、次に紹介する例のプログラムと例文を「メモ帳」(Mac「メモ」)アプリでタイプして作り、それをデスクトップ上に**指定の名前**で保存してもいいです

# 何するの？

# このプログラムを動かしてみよう

## 文字頻度を数えるRuby プログラム

[hindo.rb](https://hindo.rb)

```
# hindo.rb
# input: mojiretsu (tatoeba angobun)
# output: kaku moji (small English letter) no hindo

code_a = 97
mojiretsu = gets.chomp
nagasa = mojiretsu.length
code_retsu = mojiretsu.unpack("C*")

hindo = Array.new(26, 0)
for i in 0..(nagasa-1)
  sa = code_retsu[i] - code_a
  #print(mojiretsu[i], " ", code_retsu[i], " ", sa, "¥n")
  if 0 <= sa && sa <= 25
    hindo[sa] = hindo[sa] + 1
    #print(hindo, "¥n")
  end
end
print(hindo, "¥n")

max = -1
max_sa = -1
for sa in 0..25
  if hindo[sa] > max
    max = hindo[sa]
    max_sa = sa
  end
end
print("max = ", max, " <-- sono ichi (moji a karano sa) = ", max_sa, "¥n")
```

この名前で保存してね。  
コピーだと空白が抜けたりするので入れ直してね。あと、¥記号はタイプし直した方がよいようです。

Mac 人へ

タイプする場合には、プログラム中の¥記号はOption Key を押しながらバックスラッシュキーを押してください

## 文字頻度って？

- ・ 文章の中に各文字(ただし, 英小文字)が現れる回数のこと
- ・ それをヒントに次のような暗号文の解読をしたいのだ

### チャレンジ暗号文

spwwz pgpcjzyp! hpwnzxp ez esp hzcwo  
zq nzxafepc dntpynp! nzxafepc dntpynp  
td l mldtd zq esp xzopcy tyqzcxletzy lyo  
nzxxfytnletzy epnsyzwzrj dfns ld mtr olel  
lylwjtdt, lt, lyo lwdz nzxafepc rlxp.  
awpldp pyuzj esp hzcwo zq nzxafepc  
dntpynp. dpp jzf lrlty le  
seea://end.n.etepns.ln.ua/ndmzzv/

c.txt

← コピペするときは  
この名前で保存してね。  
ただ, コピペだと空白が  
抜けたりするので注意。  
この表示の都合上行が  
かわってますが, 改行  
は無しです。一行の  
長〜い文です。

- ・ これはシーザー暗号という暗号方式によって暗号化された英文

各英小文字をアルファベット上で  $k$  文字 ( $k$  は秘密!) 先にずらして  
得られる暗号。解読にはホームズの「踊る人形」を読もう!

次ページからの手順でやってみよう!

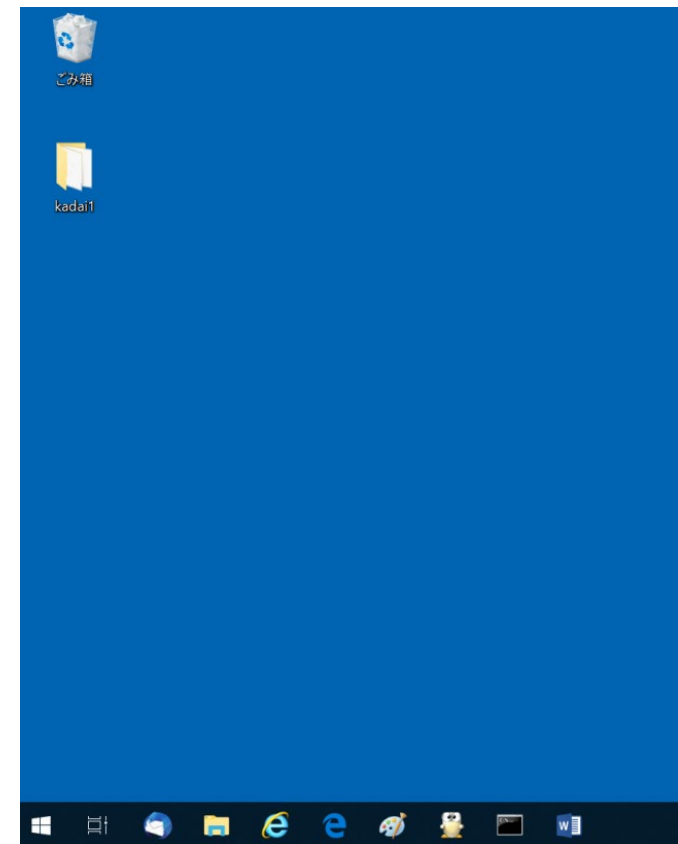
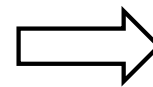
# ステップ0. デスクトップモードに変える

(1) 初期画面から → の所をクリックしてデスクトップモードにする.



「Desktop」をクリック

こんな画面に  
(背景色はPCの設定で異なる)



Mac 人へ

ログインしたら自動的にデスクトップの世界です

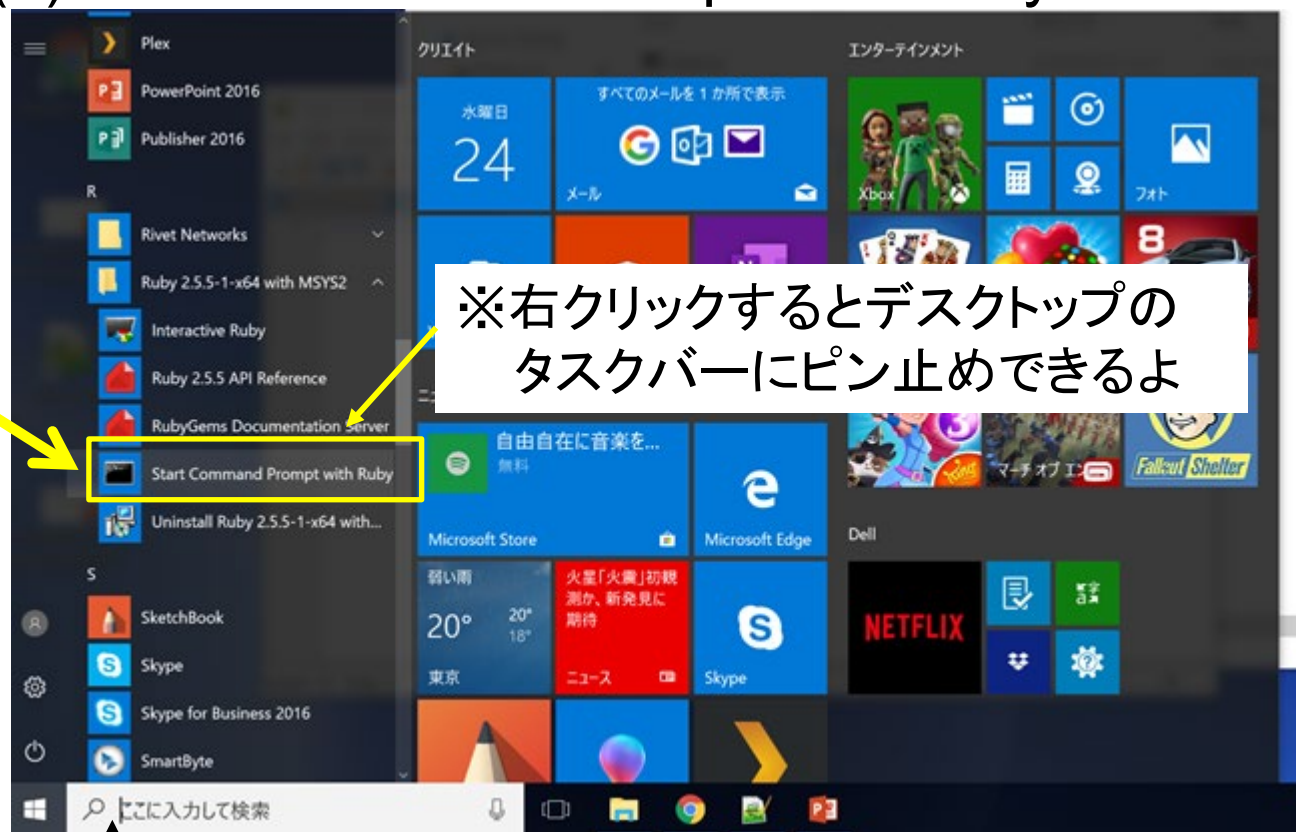
# ステップ1. 「コマンドプロンプト」を起動

Mac 人へ

「ターミナル」アプリを起動. アプリの中から探そう

(1) 画面左隅の  をリック

(2) Start Command Prompt with Ruby をクリック



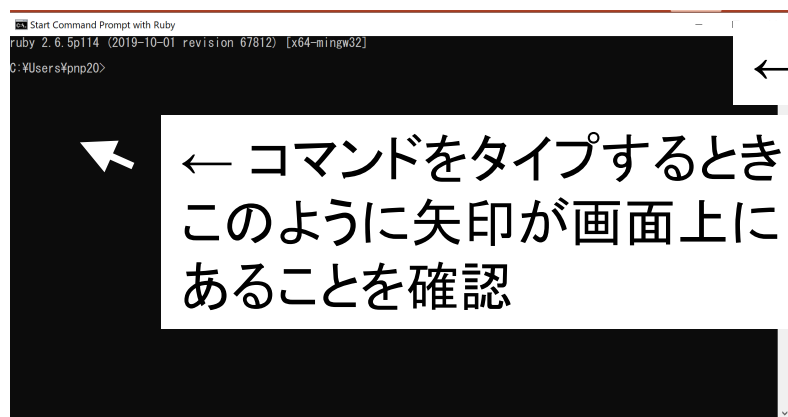
見つからない時はここに Start Command Prompt with Ruby とタイプしよう

# ステップ2. コマンドプロンプト上での操作

Mac 人へ

Mac では, ターミナルと呼ばれる

## ↓ コマンドプロンプトの実際の画面の例

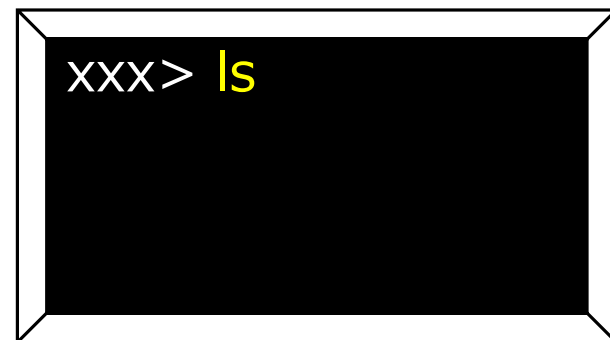


← 画面の背景色は変えられるよ

← コマンドをタイプするときは  
このように矢印が画面上に  
あることを確認

- この画面上でいろいろな**コマンド**(命令)をPC に対して指示することができる

※説明では, こんな絵を使います



↑  
本当は全部白字だけど  
区別のために, 人がタイプ  
するのは黄色にします

## ステップ2. コマンドプロンプト上での操作

Mac 人へ

次のページを見てください

※命令の最後は Return キーを忘れずに

(1) デスクトップという「お部屋」に進む: **cd Desktop**

(2) その「お部屋」の中のファイルを確認: **dir**

※ここでは、**ディレクトリ**(=**フォルダ**)を「お部屋」と呼んでいます

xxx> **cd Desktop** ← (1) これで Desktop というお部屋に移動

xxx> **dir** ← (2) 以下のような情報が表示される

ドライブ C のボリュームラベルは XXXX

ボリューム シリアル番号は ...

C:\xxx¥Desktop のディレクトリ

2020/03/05 14:30 <DIR> .

2020/03/05 14:30 <DIR?> ..

2020/03/03 19:00 hindo.rb

2020/03/03/ 19:00 c.txt

...

← Desktop というお部屋の中にあるファイル

## ステップ2. ターミナル上での操作 (Mac 用説明)

※命令の最後は Return キーを忘れずに

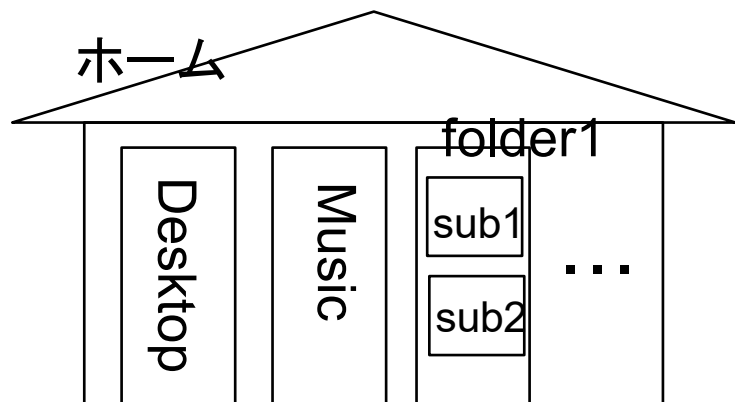
(1) デスクトップという「お部屋」に進む: **cd Desktop**

(2) その「お部屋」の中のファイルを確認: **ls**

```
xxx> cd Desktop ← (1) これで Desktop というお部屋に移動
xxx> ls ← (2)
    hindo.rb  c.txt  ...
```

← Desktop というお部屋の中にあるファイル

※ここでは、**ディレクトリ (= フォルダ)**を「お部屋」と呼んでいます



補足: 「お部屋」の構造

- ・普通は**ホーム**と呼ばれるお部屋が割り振られる
- ・その中にいくつかのお部屋があり, その1つがデスクトップ Desktop というお部屋
- ・お部屋の中に, またお部屋を作ることができる
- ・上位のお部屋に行くには **cd ..**
- ・今のお部屋の位置付けを知るには **pwd**



# ステップ3. いよいよプログラムを動かそう！

```
xxx> cd Desktop
```

```
xxx> dir
```

ドライブ C のボリュームラベルは xxxx  
ボリューム シリアル番号は ...

C:\xxx\Desktop のディレクトリ

```
2020/03/05  14:30  <DIR>      .  
2020/03/05  14:30  <DIR?    ..  
2020/03/03   19:00  hindo.rb  
2020/03/03/  19:00  c.txt
```

...

```
xxx> ruby hindo.rb  
aabbbABC
```

```
[2, 3, 0, 0, ...
```

```
max = 3 <- sono ichi (moji a kara no sa) = 1
```

← ここまではステップ2

← この 2 つのファイルが重要

← 何か文字列を入れるのを待っているので  
適当な文字列を入れる

次に続くよ

## ステップ3. いよいよプログラムを動かそう！

```
xxx> ruby hindo.rb  
aabbbbABC
```

白字が計算結果(英小文字のみ)

```
[2, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
```

↑ 文字 a が 2 回, b が 3 回, 他が 0 回という結果を表している

```
max = 3 <- sono ichi (moji a kara no sa) = 1
```

↑ 頻度 3 回が最大

↑ その文字の位置(文字 a からの距離)は 1

```
xxx> ruby hindo.rb
```

```
aaaccczzzzz
```

```
[3, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5]
```

```
max = 5 <- sono ichi (moji a kara no sa) = 25
```

```
xxx> ruby hindo.rb < c.txt
```

↑ こうやると文字列をキーボードからではなく  
ファイルから読み込む



さあて、どうなるかな？

そして秘密の鍵  $k$  を求めよう！

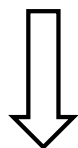
## ステップ4. 鍵がわかったら解読だ！

秘密の鍵(シフト数)  $k$  の予想ができたかな？

そうしたらその数を復号プログラムに書き入れて実行だ。

hukugo.rb

ここを予想したシフト数を書き込む



そして実行

xxx> ruby hukugo.rb < c.txt

うまくいったかな？

```
# hukugo.rb
# input: angobun (Caesar angok (k shift))
# output: hirabun (= moto no bun)

#===== koko ha kansuu (subroutine tomo iu) no teigi =====
def dec(k, c)
  code_a = 97
  nagasa = c.length
  a = c.unpack("C*")
  b = Array.new(nagasa)
  for i in 0..(nagasa-1)
    sa = a[i] - code_a
    if 0 <= sa && sa <= 25
      b[i] = code_a + ((sa - k)%26) # korede -k shift ga dekiru
    else
      b[i] = a[i]
    end
  end
  m = b.pack("C*")
  return m
end

#===== kokokara program hontai =====
angobun = gets.chomp
hirabun = dec(0, angobun)
puts(hirabun)
```

仕組みについては

<http://tcs.c.titech.ac.jp/csbook/> や

youtube で

「渡辺治 東工大」で検索

# 補足: Windows と Mac の違いについて

## 1. Windows とは, Mac とは？

Windows とか Mac というのは OS(オーエス)の略称. コンピュータ上でソフトを動かすための枠組みのこと. 正確には, たとえば Windows10 とか, macOS Mojave など, 版がことなるものがある. なお, Windows は, 多数のコンピュータで使用されているが, Mac はアップル社のコンピュータ Macintosh 上でのみ使われている.

## 2. 基本用語対応表

Windows と Mac でよく使われている基本用語の対応を以下に示す.

| Windows                     | Mac                 | 解 説   |
|-----------------------------|---------------------|---|
| フォルダ<br>folder              | ディレクトリ<br>directory | ファイルを入れておく場所. 本講義では「お部屋」と呼んでいる. Windows, Mac で両方の呼び方を使う場合もある. |
| ホーム<br>home                 | ホーム<br>home         | アプリの拠点となる「お部屋」(アプリごとに異なる).                                    |
| エクスプローラー                    | ファインダー              | ディレクトリの中を見たりファイルを移動させる時に使うソフト. Windows ではフォルダを開くと自動的に使われる.    |
| コマンドプロンプト<br>Command Prompt | ターミナル<br>Terminal   | コンピュータへの命令(コマンド)を入力して実行するための場所. 一般的にはシェルと呼ばれている.              |

## 補足: Windows と Mac の違いについて

### 3. コマンドプロンプト上の基本命令

以下は, コマンドプロンプト (Mac ではターミナル) 上でよく使われる基本命令.  
シルバー字で括弧に入っているのが Mac での名称. 以下のように, 基本命令では `dir` と `ls` しか変わらないので, その点だけ注意すれば大差なく使える.

| 命令                    | 使用例                       | 意 味                                |
|-----------------------|---------------------------|------------------------------------|
| <code>mkdir</code>    | <code>mkdir kadai2</code> | kadai2 という部屋 (フォルダ (ディレクトリ)) を作る   |
| <code>cd</code>       | <code>cd kadai2</code>    | kadai2 という部屋に入る                    |
|                       | <code>cd ..</code>        | 上の (大きな) 部屋に戻る                     |
|                       | <code>cd ../..</code>     | 上の上の部屋に戻る                          |
| <code>dir (ls)</code> | <code>dir (ls)</code>     | その部屋にあるファイルを表示する                   |
| <code>more</code>     | <code>more foo.txt</code> | ファイル <code>foo.txt</code> の中身を見る   |
| <code>rm</code>       | <code>rm foo.rb</code>    | <code>foo.rb</code> を消す (戻らないので注意) |

## (2) ターミナル上の基本命令について

### 1. ターミナルとは、そもそも、Mac とは？

Windows とか Mac というのは OS(オーエス)の略称. コンピュータ上でソフトを動かすための枠組みのこと. 正確には, たとえば Windows10 とか, macOS Mojave など, 版がことなるものがある. なお, Windows は, 多数のコンピュータで使用されているが, Mac はアップル社のコンピュータ Macintosh 上でのみ使われている. ターミナル(Terminal)とは, コンピュータへの命令(コマンド)を入力して実行するための場所のこと. 一般的にはシェルと呼ばれている.

### 2. ターミナル上の基本命令

実習でも説明するので, ここでは, ごく基本の命令についてまとめておく.

| 命令    | 使用例          | 意 味                  |
|-------|--------------|----------------------|
| mkdir | mkdir kadai2 | kadai2 という部屋※を作る     |
| cd    | cd kadai2    | kadai2 という部屋に入る      |
|       | cd ..        | 上の(大きな)部屋に戻る         |
|       | cd ../..     | 上の上の部屋に戻る            |
| ls    | ls           | その部屋にあるファイルを表示する     |
| rm    | rm foo.rb    | foo.rb を消す(戻らないので注意) |

※ フォルダとかディレクトリと呼ばれるもの. 本講義では比喩的に「部屋」と呼んでいる.