18 Python基礎

condaの環境の整備

condaのアップデート

- 使用中の仮想環境の更新（やや遅い）

```
> conda activate py39
> conda update --all
```

- たまに以下の表示がされる場合がある

```
Please update conda by running

    $ conda update -n base -c defaults conda
```

condaコマンド自体の更新！（結構遅い）

```
> conda update -n base -c defaults conda
```

pythonプログラムのexe化

- pyinstallerのインストール

```
> conda install pyinstaller
```

- pyinstallerを利用して、.py→.exeに変換

```
> pyinstaller hoge.py --onefile
```

- hoge.pyと同じディレクトリ内にdistが生成され、中にhoge.exeが作られる。
    - 単独で使用可能な実行ファイルが生成される。

wc.pyのexe化

- 前回作成したwc.pyを実行ファイルにし、動作確認を行ってください。

```
(py39) Z:\Dropbox\2024Trident\zenki\講義資料\NT_Python基礎\src\18>tree ./
フォルダー パスの一覧:  ボリューム VBOX_yoshimura
ボリューム シリアル番号は 0000008D 0001:0303 です
Z:\DROPBOX\2024TRIDENT\ZENKI\講義資料\NT_PYTHON基礎\SRC\18
├─dist
└─build
    └─wc
        └─localpycs
```

- dist内にwc.exeが生成される。
- PATHを指定して実行する

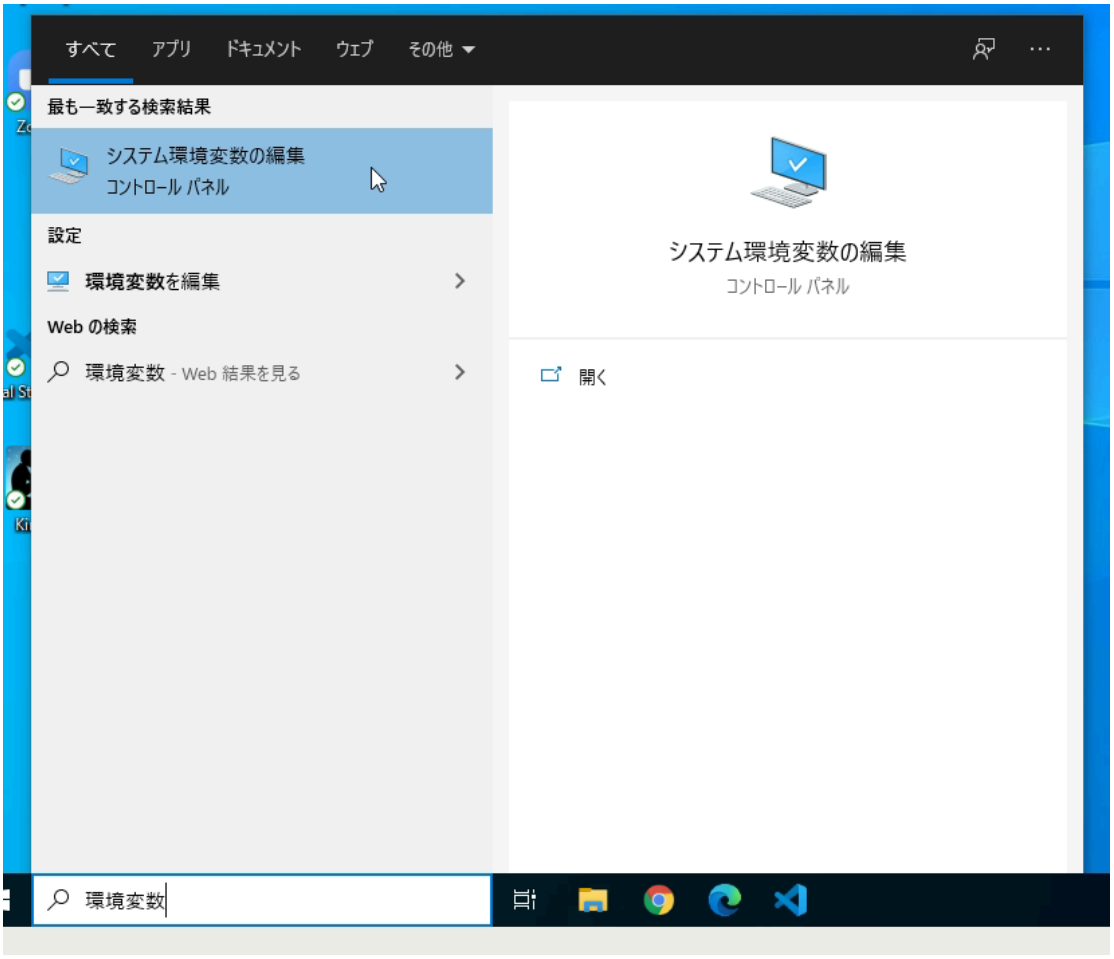
```
> dist\wc.exe wc.py
208 640 7181 wc.py
```

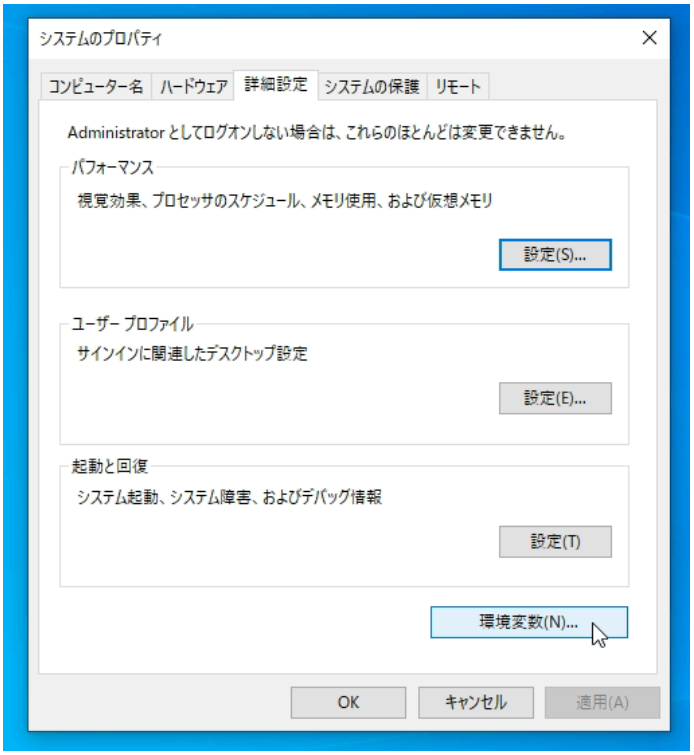
- busyboxを導入している場合、PATHを指定しないとbusybox版が動く

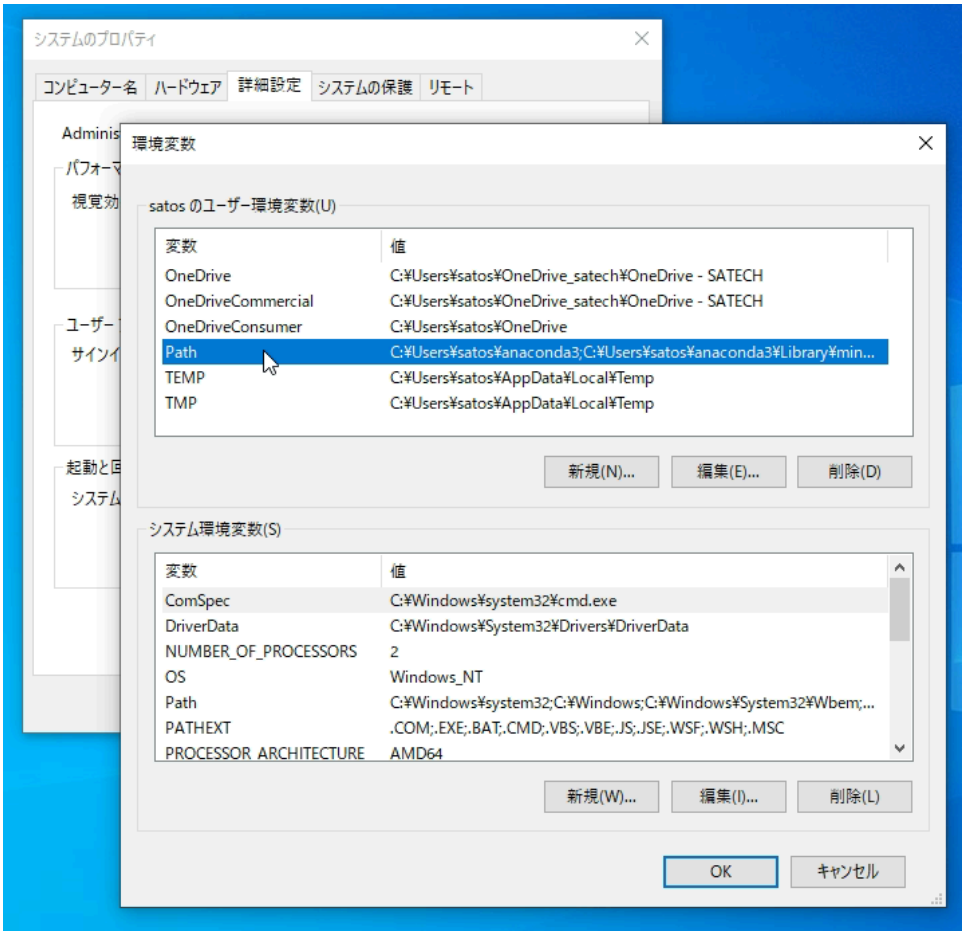
※ pyinstallerを使用してexe化した場合の問題点を考えてください。

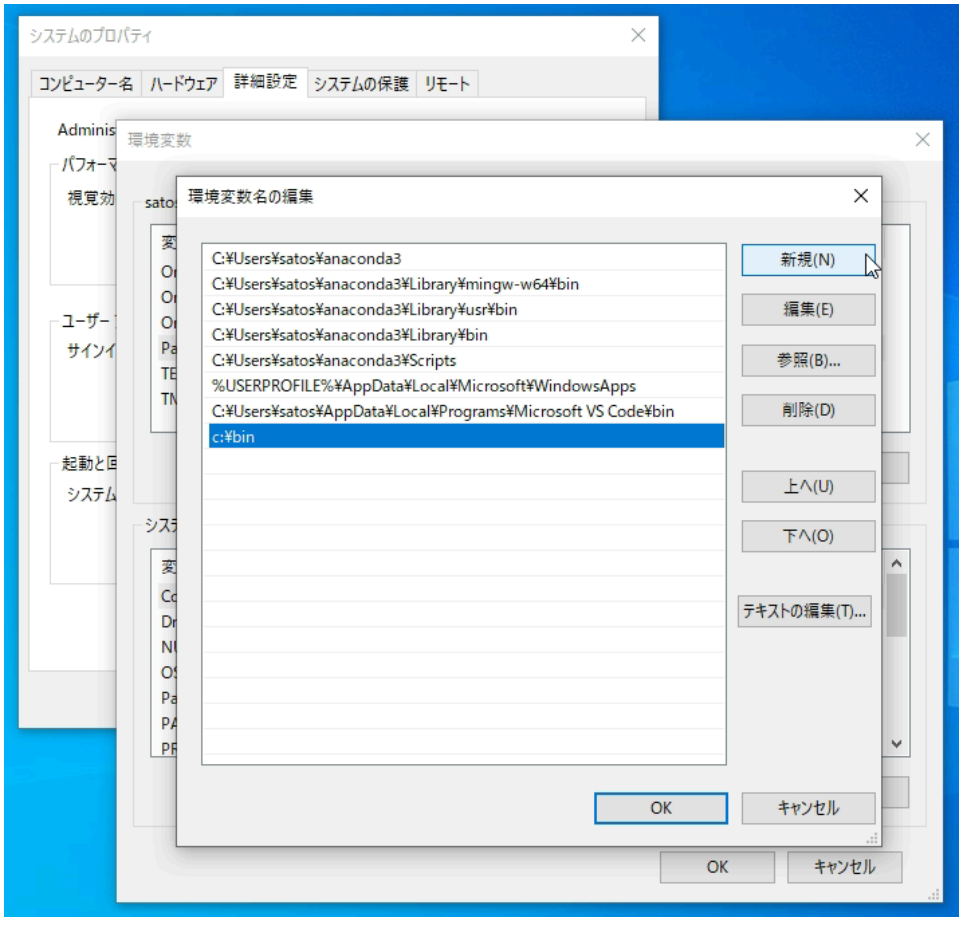
pathの設定と配置

- 自分でさくせしたTool類（.exe）を特定のフォルダに配置し、使えるようにする。
    - 例として、C:\binに格納する。
- 環境変数PATHに上記フォルダを登録する。


すべて
アプリ
ドキュメント
ウェブ
その他
最も一致する検索結果
システム環境変数の編集
コントロール パネル
設定
環境変数を編集
Web の検索
環境変数 - Web 結果を見る
システム環境変数の編集
コントロール パネル
開く
環境変数


システムのプロパティ
コンピューター名
ハードウェア
詳細設定
システムの保護
リモート
Administrator としてログオンしない場合は、これらのほとんどは変更できません。
パフォーマンス
視覚効果、プロセッサのスケジュール、メモリ使用、および仮想メモリ
設定(S)...
ユーザー プロファイル
サインインに関連したデスクトップ設定
設定(E)...
起動と回復
システム起動、システム障害、およびデバッグ情報
設定(T)
環境変数(N)...
OK
キャンセル
適用(A)





- ユーザの環境変数→PATHの編集→新規で「C:¥bin」を登録する。

- Windowsは、PATHに登録されたフォルダ内から指定されたファイル名の実行ファイルを探し、実行する。
    - 上記設定により、コマンドプロンプトで、hogeと入力することで、hoge.exeを実行することが可能になる。
    - PATHの上の方にあるフォルダから実行するため、優先順位を上げる場合は「上」の方に設定する
        - 同じ名前の実行ファイルが存在する場合、上位のフォルダから実行される。
        - 吉村は最上位に「c:\bin」を登録している

正規表現

- 正規表現(Regular Expression)モジュール `re` を使用する

  > https://docs.python.org/ja/3.9/library/re.html

  - 各メソッドについて、良く読んで理解してください。

正規表現を覚えているだろうか？

- ワイルドカード `*` や `?` を拡張（高機能化）したもの
  - 今一度、復習してください。

    > https://userweb.mnet.ne.jp/nakama/
    >
    > https://zenn.dev/ryome/articles/9a28660d27363b

練習1

- 正規表現を使用して以下のプログラムを作成してください。

1. `Yahoo_index.html`（2025-06-18保存データ）内に含まれる、条件に当てはまる文字列を探し出して出力してください。（re1_1.py）
   - 条件
     - ダブルクォートで囲まれた文字列で、`http://` もしくは `https://` から始まる文字列（URLらしきもの）を取り出す
       - QueryStringがある場合（以下のような場合）は、QueryStringを含めて1つとする。

         ```
         https://www.hoge.jp/?a=10&b=20
         ```

     - 上記のデータが何件（何個）あるか確認してください。
       【実行例】

```
$ python ren1_1.py
0 - https://m.yahoo.co.jp/
1 - https://www.yahoo.co.jp/
2 - https://s.yimg.jp/c/icon/s/bsc/2.0/favicon.ico
3 - https://s.yimg.jp/c/icon/s/bsc/2.0/favicon.ico
4 - https://s.yimg.jp/c/icon/s/bsc/2.0/y120.png
5 - https://www.yahoo.co.jp/
6 - https://s.yimg.jp/images/top/ogp/fb_y_1500px.png
7 - https://s.yimg.jp/images/top/ogp/tw_y_1400px.png
8 - https://s.yimg.jp/images/ds/managed/1/managed-ual.min.js?tk=4465a92c-f0fd-406f-b519-efd409cc9849&amp;service=toppage
9 - https://yads.c.yimg.jp/js/yads-async.js
  ⋮
```

2. ren1_1.pyの結果を集計したい（ren1_2.py）

- 各データの内容を、同じデータ毎に何個あるかカウントする。
- データ数の多い方から出力する。
  - ただし、個数が1個のものは出力しないものとする。
    【実行結果】

```
2 - https://www.yahoo.co.jp/
2 - https://s.yimg.jp/c/icon/s/bsc/2.0/favicon.ico
  ⋮
```

3. 他のファイルでも確認してたい（ren1_3.py）

- 複数のファイル名をリストにもたせて、ループでファイルごとの結果を出力させる。
- 出力件数は、各ファイルごとに最大10件とする

1. Yahoo.html
2. Internet.html
3. Ascii.html
4. Forest.html

4. ファイル名を入力するのは手間なので、特定のフォルダ内のファイルを全て処理するようにする（ren1_4.py）

- htmlフォルダを作成し、その中に調べたいhtmlを入れる
- htmlフォルダ内の全てのhtmlに対して、3の処理を行う

5. フォルダ名を固定にするのは、不便なので、コマンドライン引数で渡せるようにする。（ren1_5.py）