

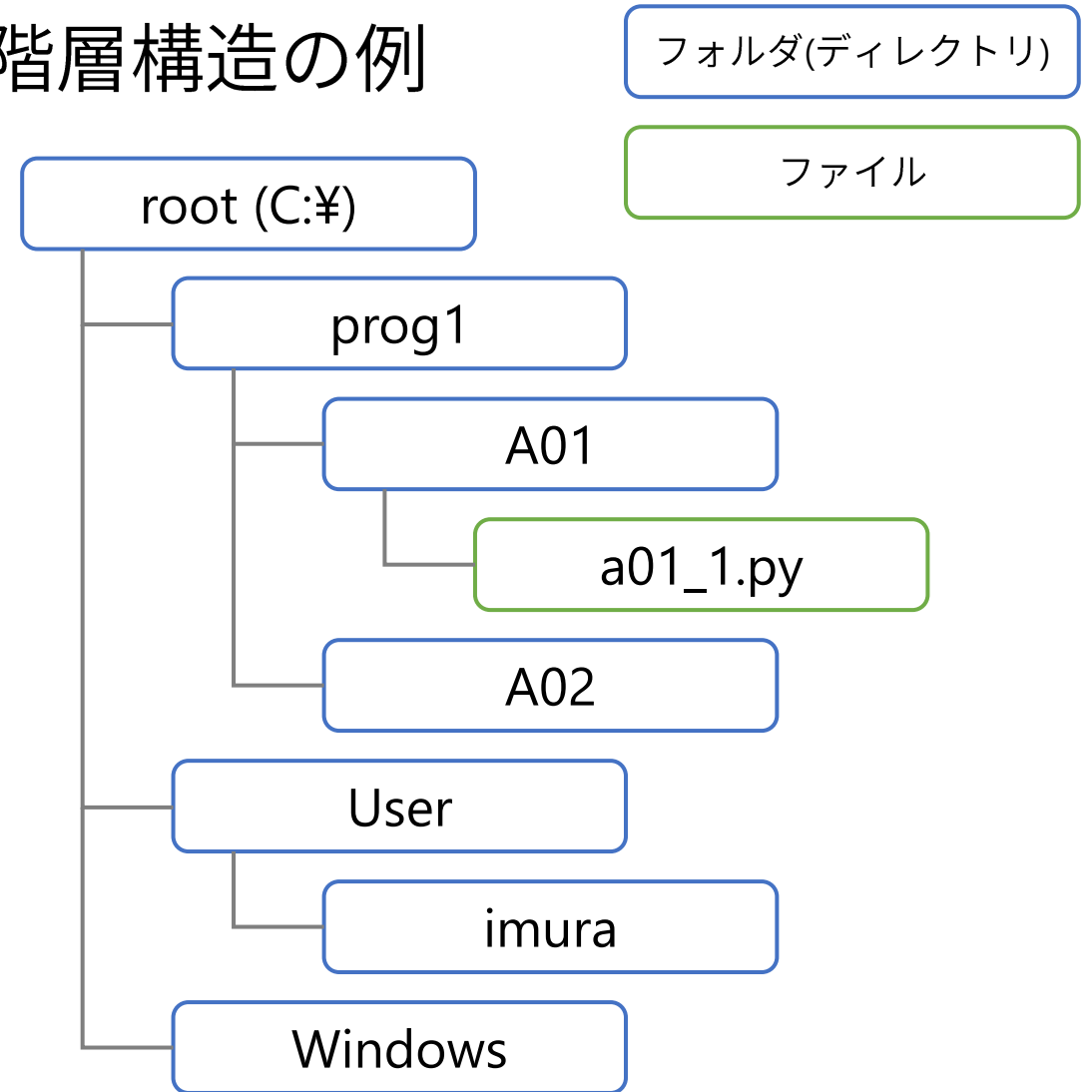
# ファイルシステムの構造とコマンドラインでの移動

---

# ファイルシステムは木構造

- コンピュータのファイルシステムは、情報を**木構造(階層構造)**で整理することが一般的
  - 大元: ルートディレクトリ
  - 途中: フォルダ(ディレクトリ)
    - フォルダとディレクトリは同じものと思ってもらってよい.
      - コマンドラインではディレクトリと呼ばれることが一般的
  - 末端: ファイル
  - フォルダあるいはファイルに至るまでにたどる経路: パス(path)

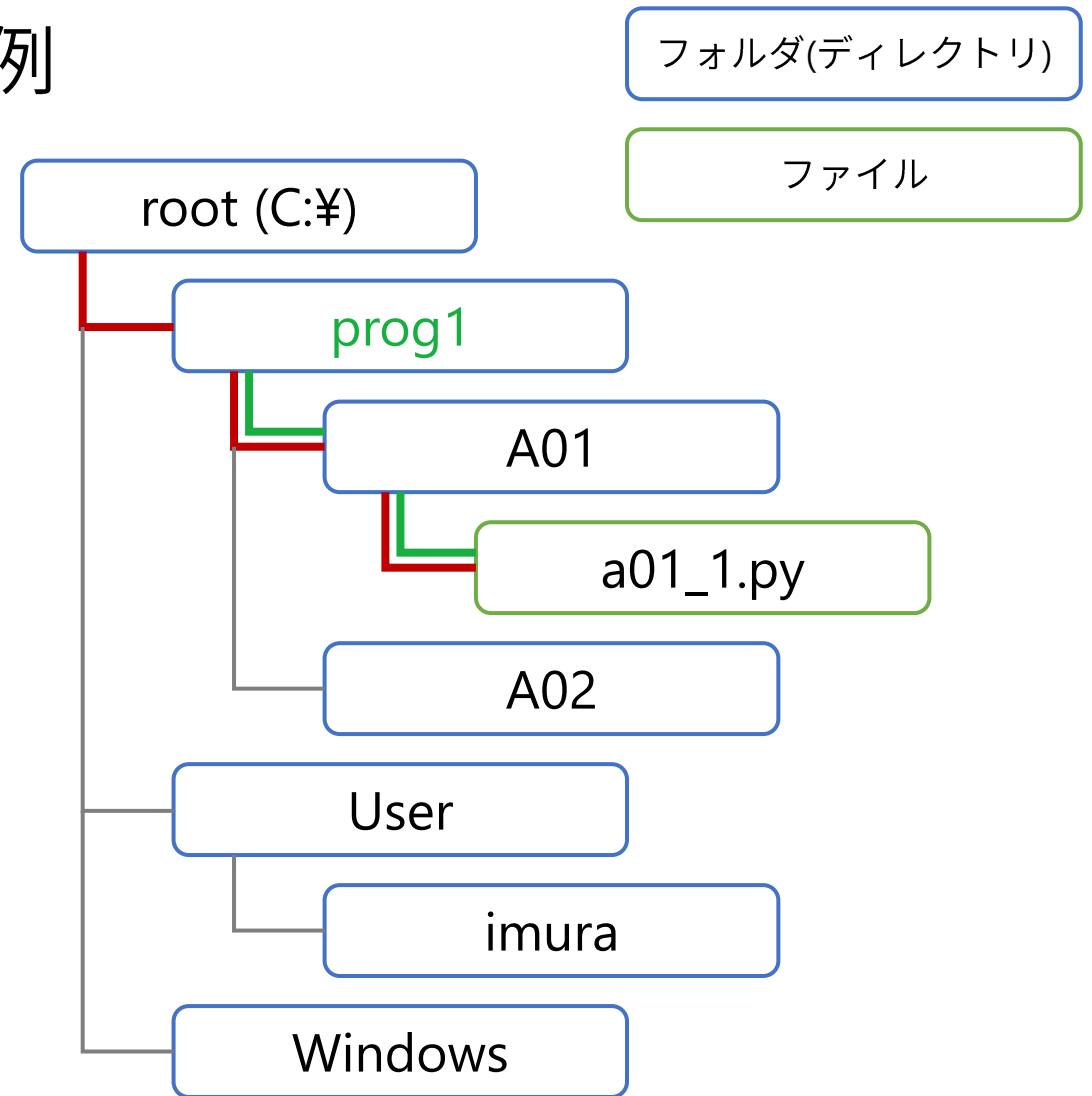
## ● 階層構造の例



# パス(path)

- フォルダ(ディレクトリ)やファイルのある場所の指定
  - フォルダを¥(Windows)か/(Windows以外)のいずれかで区切って表す
- 絶対パス: ルートからのパス
  - 最初が ¥ ではじまる
  - 例 `C:¥prog1¥A01¥a01_1.py`
- 相対パス: カレントディレクトリ(今いるところ)からのパス
  - 最初が ¥ 以外
  - 例 `prog1にいるとき, A01¥a01_1.py`

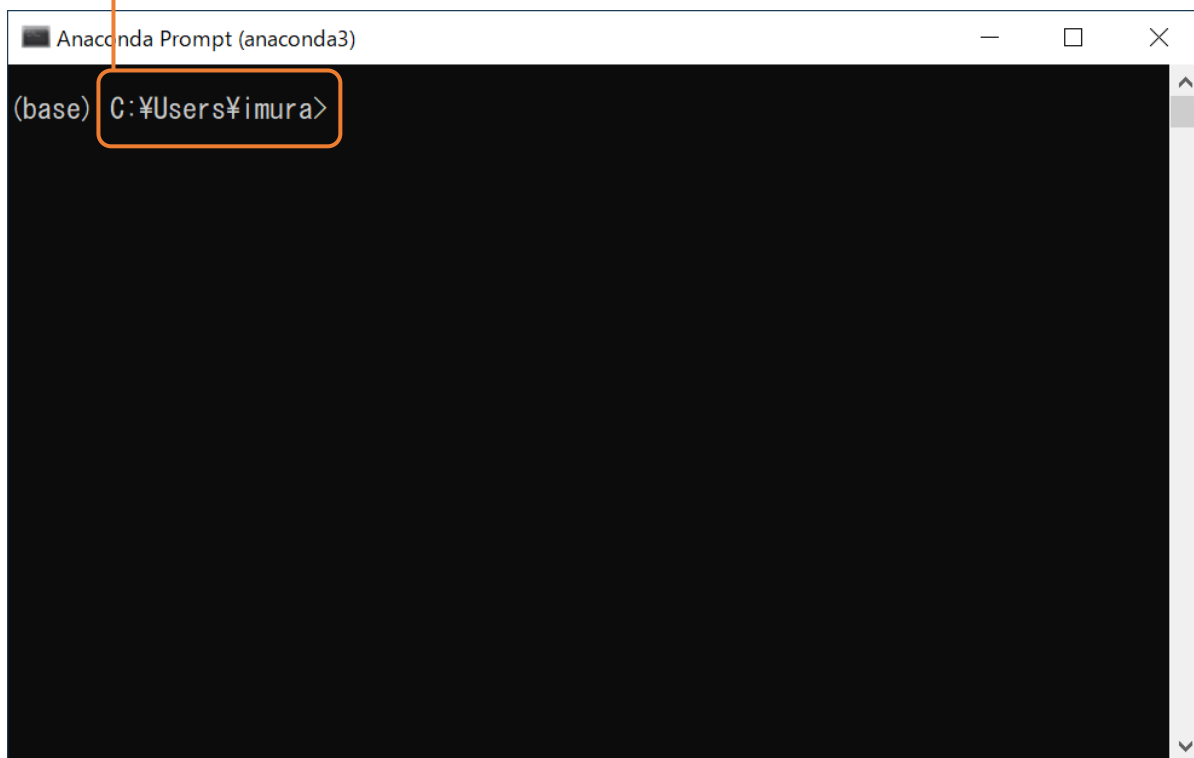
## ● 例



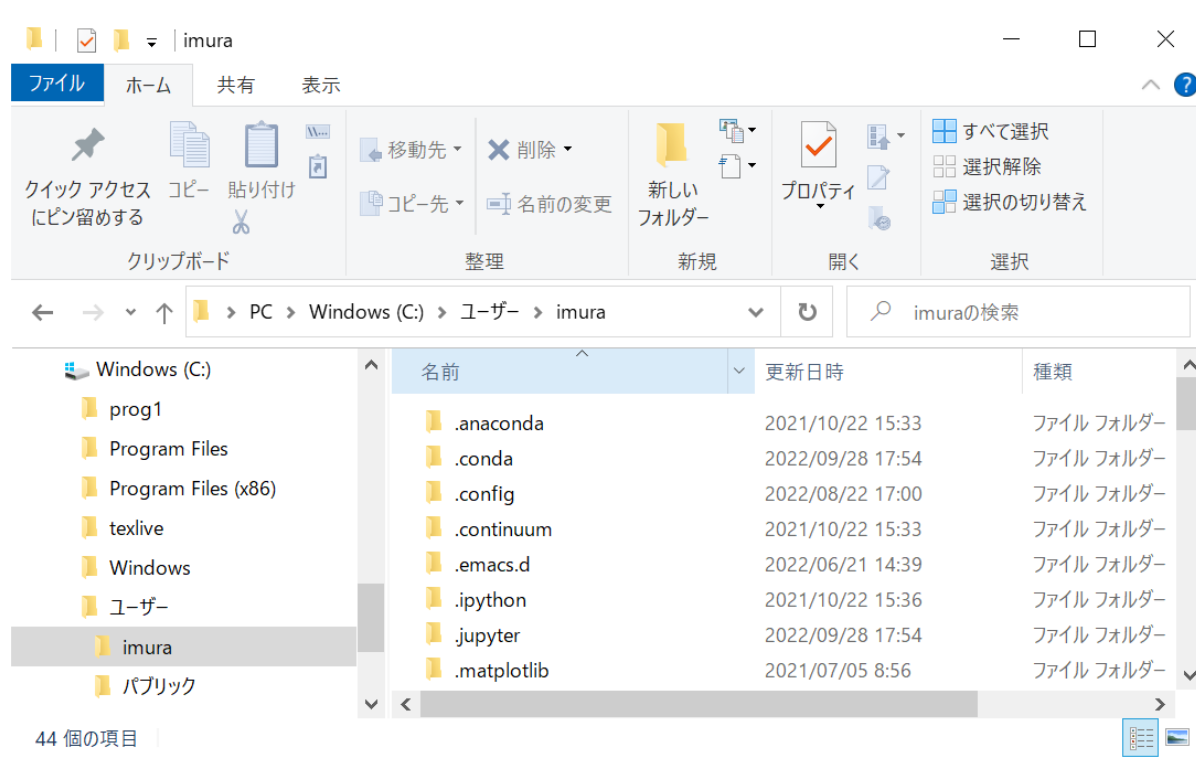
# コマンドラインとエクスプローラーを並べて確認

## カレントディレクトリ

明示的にディレクトリを指定しなければ、このディレクトリ内にあるファイルやディレクトリのみ操作可能



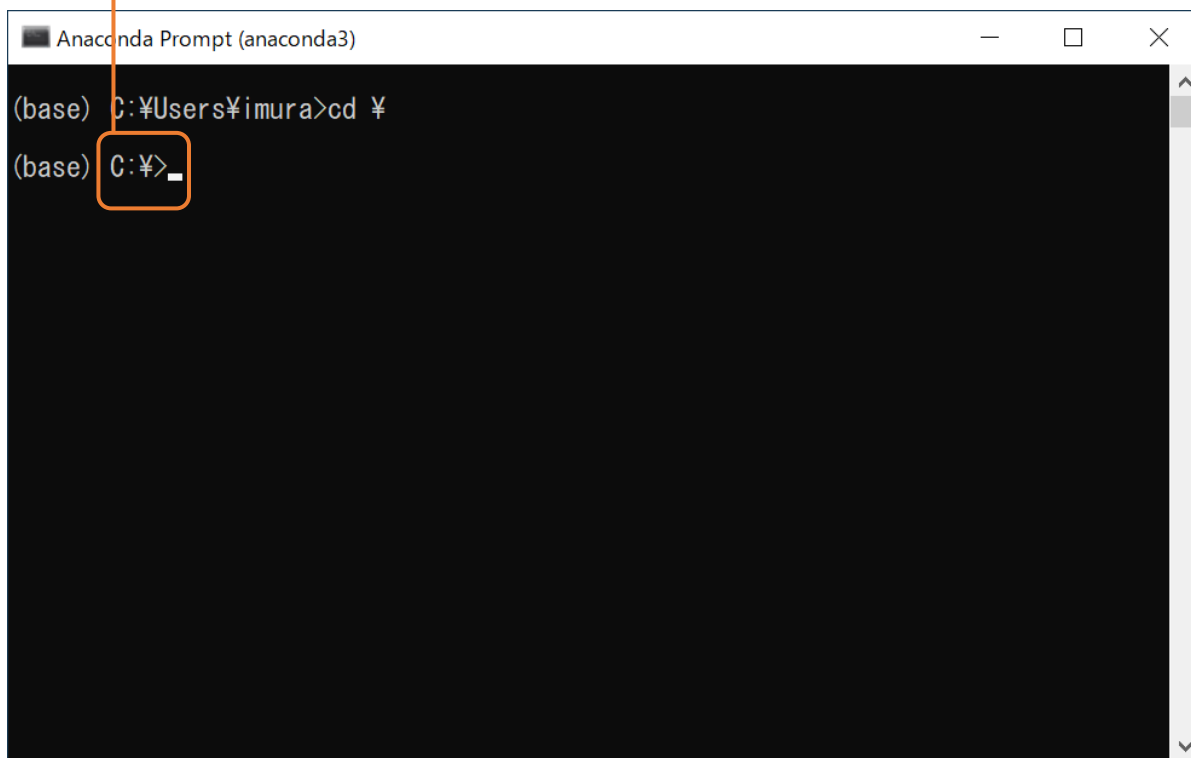
```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>
```



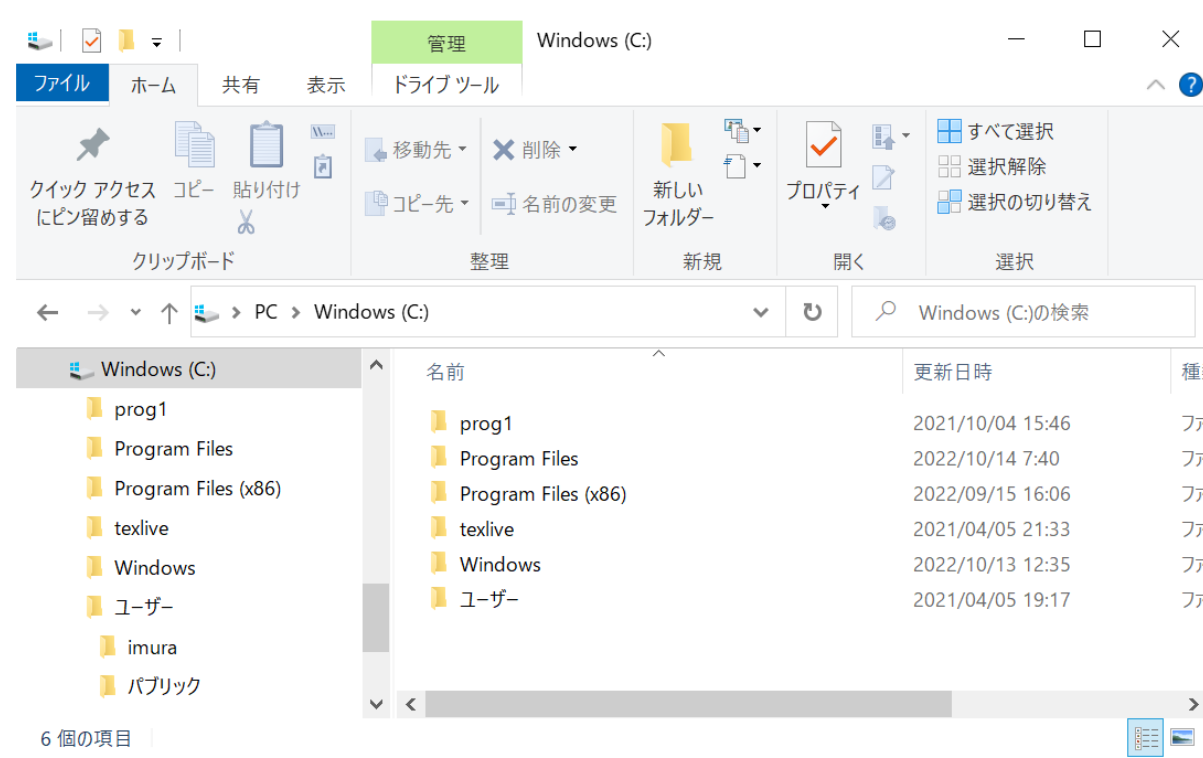
見た目は異なりますが、実体は一つ(同一のファイルシステム)

# コマンドラインとエクスプローラーを並べて確認

カレントディレクトリが変化していることに注意



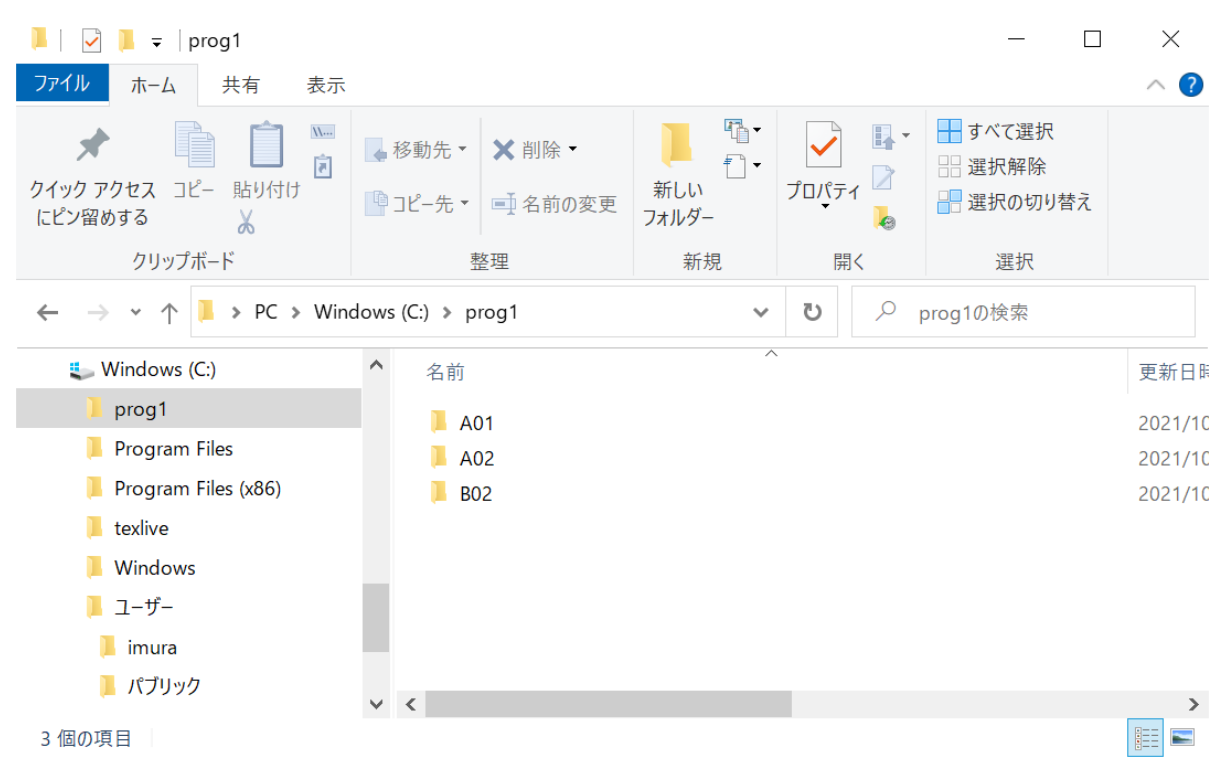
```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>cd ¥
(base) C:\>_
```



C:ドライブのルートディレクトリに移動

# コマンドラインとエクスプローラーを並べて確認

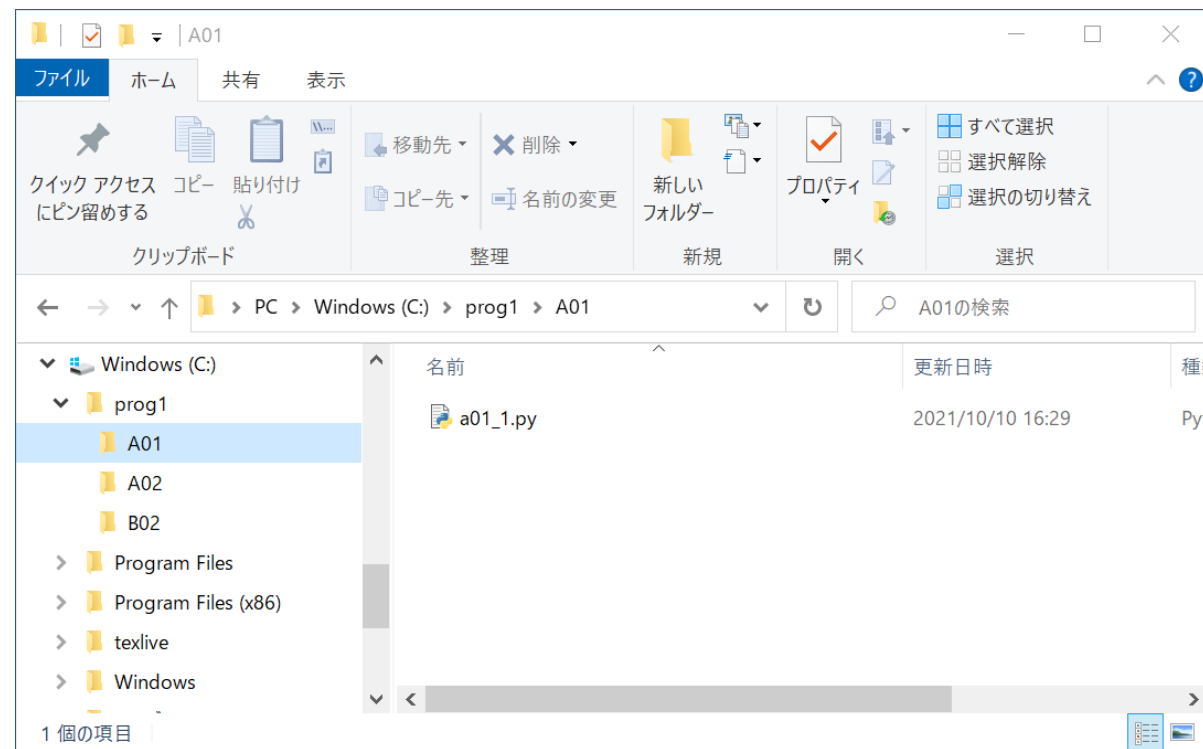
```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>cd ¥
(base) C:\¥>cd prog1
(base) C:\¥prog1>_
```



ディレクトリprog1に移動(相対パス指定)

# コマンドラインとエクスプローラーを並べて確認

```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>cd ¥
(base) C:\¥>cd prog1
(base) C:\¥prog1>cd A01
(base) C:\¥prog1¥A01>
```



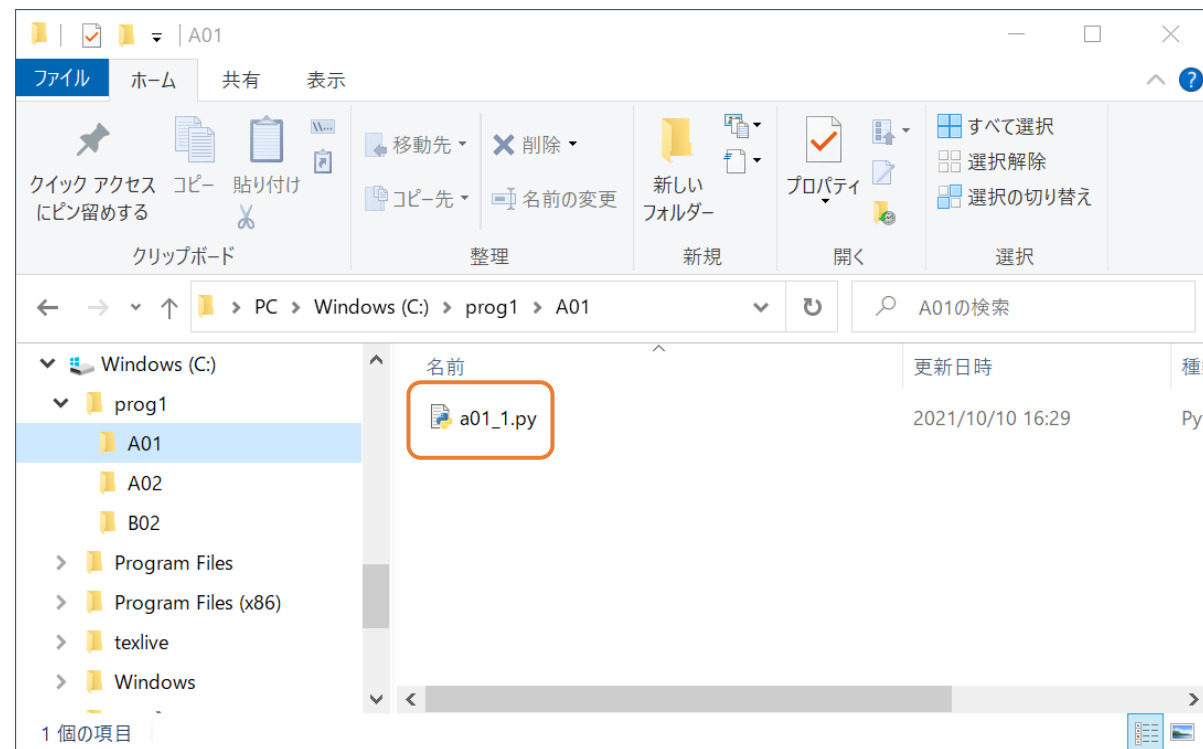
ディレクトリA01に移動(相対パス指定)

# コマンドラインとエクスプローラーを並べて確認

```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>cd ¥
(base) C:¥>cd prog1
(base) C:¥prog1>cd A01
(base) C:¥prog1¥A01>dir
ドライブ C のボリューム ラベルは Windows です
ボリューム シリアル番号は B299-0527 です

C:¥prog1¥A01 のディレクトリ
2022/10/14 09:10 <DIR>
2022/10/14 09:10 <DIR>
2022/10/14 09:12          24 a01_1.py
                1 個のファイル
                2 個のディレクトリ 259,400,912,896 バイトの空き領域

(base) C:¥prog1¥A01>
```



内容(存在しているファイル)が同一であることを確認

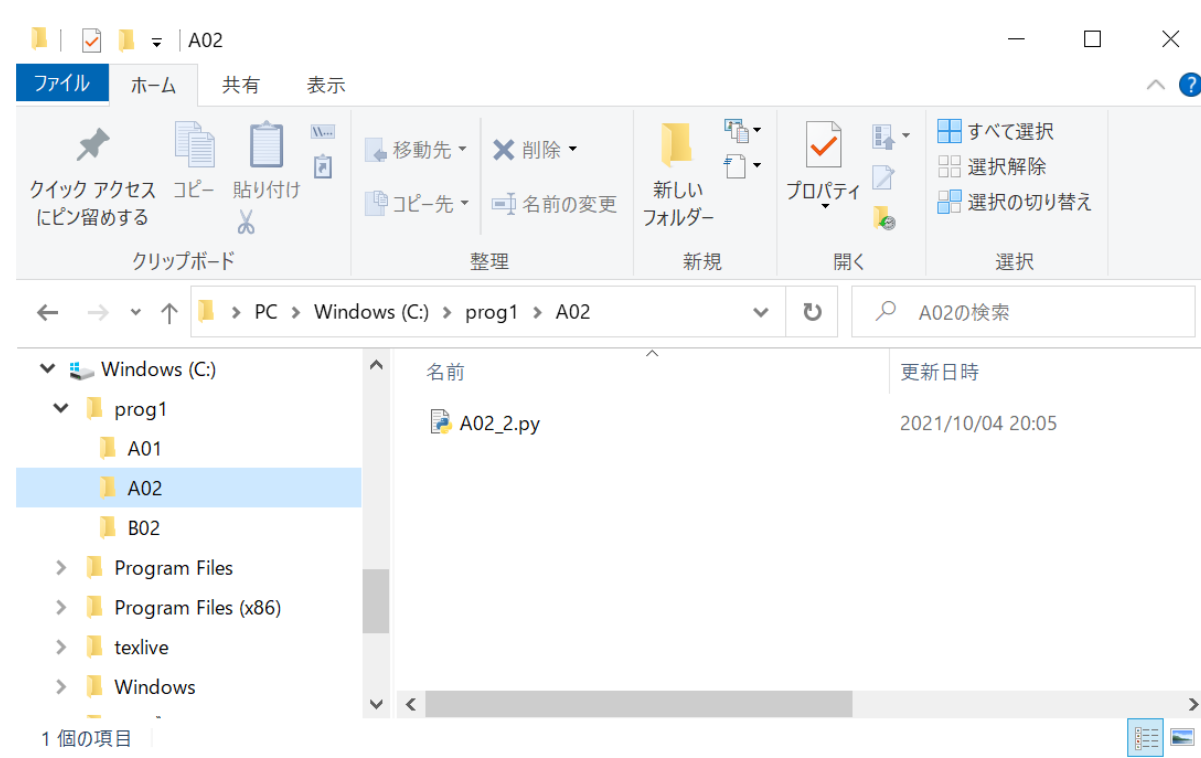


# コマンドラインとエクスプローラーを並べて確認

```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>cd ¥
(base) C:\>cd prog1
(base) C:\prog1>cd A01
(base) C:\prog1\A01>dir
ドライブ C のボリューム ラベルは Windows です
ボリューム シリアル番号は B299-0527 です

C:\prog1\A01 のディレクトリ
2022/10/14  09:10    <DIR>
2022/10/14  09:10    <DIR>
2022/10/14  09:12                24 a01_1.py
                1 個のファイル                24 バイト
                2 個のディレクトリ 259,400,912,896 バイトの空き領域

(base) C:\prog1\A01>cd ..\A02
(base) C:\prog1\A02>
```



ディレクトリA02に移動(相対パス指定)

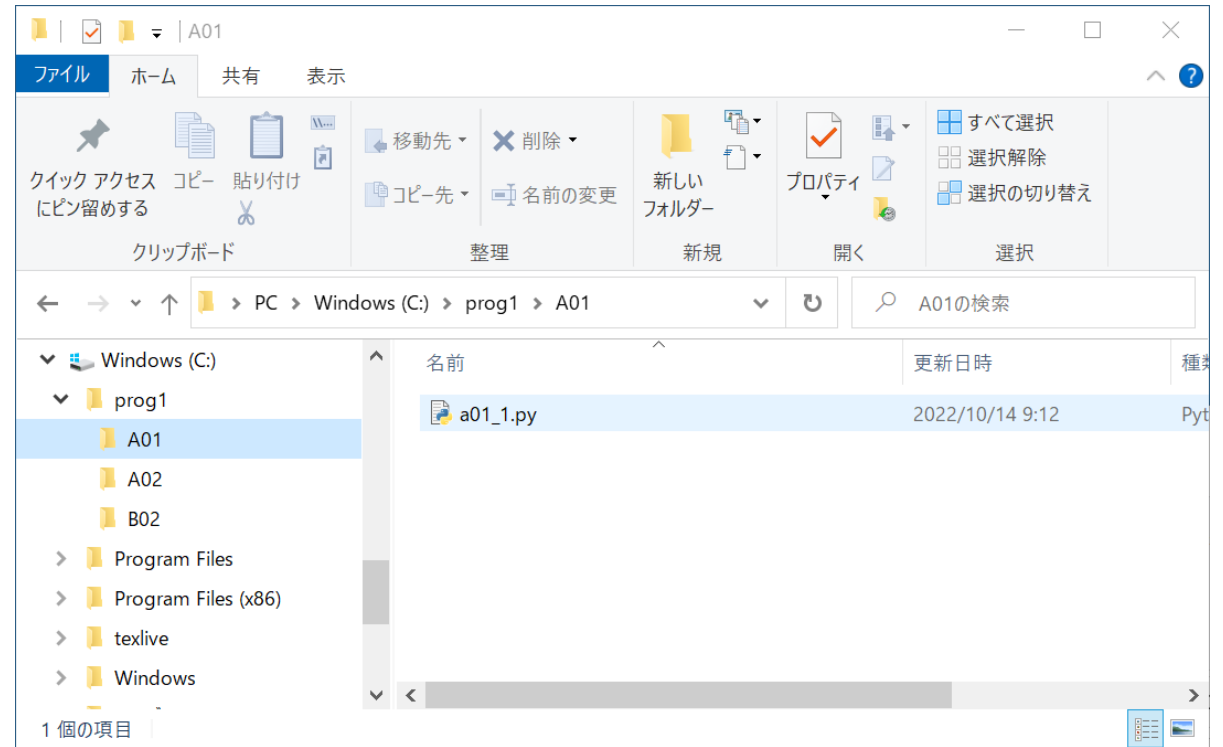
# 絶対パス指定の例

```
Anaconda Prompt (anaconda3)
(base) C:\Users\imura>cd prog1\A01
(base) C:\prog1\A01>dir
ドライブ C のボリューム ラベルは Windows です
ボリューム シリアル番号は B299-0527 です

C:\prog1\A01 のディレクトリ

2022/10/14 09:10 <DIR>
2022/10/14 09:10 <DIR>
2022/10/14 09:12          24 a01_1.py
          1 個のファイル                24 バイト
          2 個のディレクトリ 259,669,348,352 バイトの空き領域

(base) C:\prog1\A01>_
```



ディレクトリA01に直接移動(絶対パス指定)

# 演習

- 1つ上の階層のフォルダに移動する
  - 例: C:\prog1\A01 → C:\prog1
- 2つ上の階層のフォルダに移動する
  - 例: C:\prog1\A01 → C:\
- 同じ階層にあるフォルダに移動する
  - 例: C:\prog1\A01 → C:\prog1\A02
- ルートディレクトリに戻る
  - 例: C:\prog1\A01 → C:\
- 違うフォルダにあるPythonスクリプトを実行する
  - 例: カレントディレクトリがC:\prog1\A02のときに, C:\prog1\A01\A01\_1.pyを実行する.

# 解答

- 1つ上の階層のフォルダに移動する

- 例: C:\prog1 → C:\prog1

- 2つ上の階層のフォルダに移動する

- 例: C:\prog1 → C:\

- 同じ階層にあるフォルダに移動する

- 例: C:\prog1 → C:\prog1\A02

- ルートディレクトリに戻る

- 例: C:\prog1 → C:\

- 違うフォルダにあるPythonスクリプトを実行する

- 例: カレントディレクトリがC:\prog1\A02のときに, C:\prog1\A01\A01\_1.pyを実行する.

```
(base) C:\prog1>cd ..
```

```
(base) C:\prog1>
```

```
(base) C:\prog1>cd ..\..
```

```
(base) C:>
```

```
(base) C:\prog1>cd ..\A02
```

```
(base) C:\prog1\A02>
```

```
(base) C:\prog1>cd \
```

```
(base) C:>
```

```
(base) C:\A02>python ..\A01\A01_1.py
```

```
hello, world!
```

```
(base) C:\A02>python \prog1\A01\A01_1.py
```

```
hello, world!
```