

Python講座

2017/6/1(木)

ファイル操作

- プログラムからファイルの中身を読み取り、書き込みを行う
- ファイルの情報を変数に入れてあげることで実行中のプログラム内で使用できるようになる
- 実行中のプログラムで求めた値をファイルに書き込むことで別のプログラムからも値を利用できる

使用例

- セーブデータの書き込みとロード
 - 学習型AIの学習経過保存
 - オンラインで離れた場所からの作業
- etc..

ファイルを開く

- `file = open(r"C:¥Users¥admin¥Desktop¥sample.txt", "r")`
- `file = open("C:¥¥Users¥admin¥¥Desktop¥¥sample.txt", "r")`
⇒ `file`が`sample.txt`として扱われる(中身が書かれる訳ではない)

⇒ ファイルを右クリック、プロパティから場所を探す
バックスラッシュ(反転した/)は¥で代替

⇒ "r"は読み込みのみを示す

モード一覧

- r : 読み込みのみ。ファイルが見つからないとエラー。
- w : 書き込みのみ。ファイルの元の内容は削除される。
ファイルが見つからないと新規作成。
- a : 追加書き込み。ファイルの元の内容は保持される。
ファイルが見つからないと新規作成。
- r+ : 読み書き両用。見つからないとエラー
- w+ : 読み書き両用。見つからないとwと同じ。
- a+ : 読み、追加書き込み両用。見つからないとaと同じ。

ファイルを閉じる

- file.close

⇒ 見た目上必要なさそうに見えるがやっておかないと後でひどいことになる…たぶん

⇒ 意図しない書き込みなどが発生する可能性がある…たぶん

ファイルの開閉を一括で

- `with open(r"C:¥Users¥admin¥Desktop¥sample.txt", "r") as file:`
 `print(file.read())`
⇒ ファイルの開閉を一括でできる。閉め忘れがない。

⇒ ただしブロック内でしか利用できないことに注意。

ファイルの読み込み

- `file.read()`
⇒ ファイルの中身すべてを文字列として読み込み。
- `file.readlines()`
⇒ ファイルの中身すべてを一行毎にリストの要素に割り当て。
- `file.readline()`
⇒ ファイルの一行だけを文字列として読み込み。

ファイルの書き込み

- `file.write(“文字列”)`
 - ⇒ 文字列をファイルに書き込み。元の中身は失われる。
- `file.writelines(文字列リスト)`
 - ⇒ ここでの文字列はリストになっていることに注意。
 - ⇒ 中身を一行ずつ書き込み。元の中身は失われる。

書き込みにおける特殊文字

一部の文字は特殊な書き方をしないと書き込みができない。

- ¥¥ : ¥そのもの
- ¥” : “ (ダブルクォーテーション)
- ¥n : 改行

etc.

Unicode

- Pythonでは文字や数字をUnicodeという形式で表わしている。
 - ⇒ 数字の0 = 48
 - ⇒ アルファベットのA = 65
 - ⇒ ひらがなのあ = 12354
- 感覚としては文字や数字ががあるリストの何番目か(コードポイント)にあり、そこから利用するという感じ。

Unicodeの変換

- `ord(“文字”)`
 - ⇒ 文字のコードポイントを取得
 - ⇒ `ord(“A”) = 65`
- `chr(数字)`
 - ⇒ その数字に対応するコードポイントの文字を取得
 - ⇒ `chr(65) = “A”`

演習問題①

- テーマ：じゃんけん

1. プレイヤーとCPUがじゃんけんを行う
2. 互いの手とプレイヤーの勝敗をファイルに記録
3. ファイルを読み込み、プレイヤーの勝率を計算できるようにする
4. グー、チョキ、パーそれぞれの勝率を計算する

演習問題②

- テーマ：ファイルの暗号化
- 題材：演習問題①のファイル
- コードポイントを利用し、そのファイルを暗号化、読み込んだ際に復元できるようにする
- 例)暗号化：すべての文字のコードポイントを+10しファイルに書き込む 復元：ファイルから読み込んだ文字のコードポイントを-10する