Python講座

2016/6/16(木)

今回の内容

- ファイル操作
- 使用例①②
- open()①②
- close()
- with文
- 読み込み
- 書き込み
- エスケープシーケンス
- Unicode①②
- 演習問題①②

※P.101~

ファイル操作

- プログラム内でファイルを指定し、その中身を読み込んだり、中身に書き込んだりすること
- プログラムを実行するときの変数の値を、ファイルから指定することで、そこに書かれている値でプログラムが実行できる
- プログラム内で求まった値をファイルに書き込むことで、プログラムが終了後もその値を別のプログラムで使用することができる

使用例①

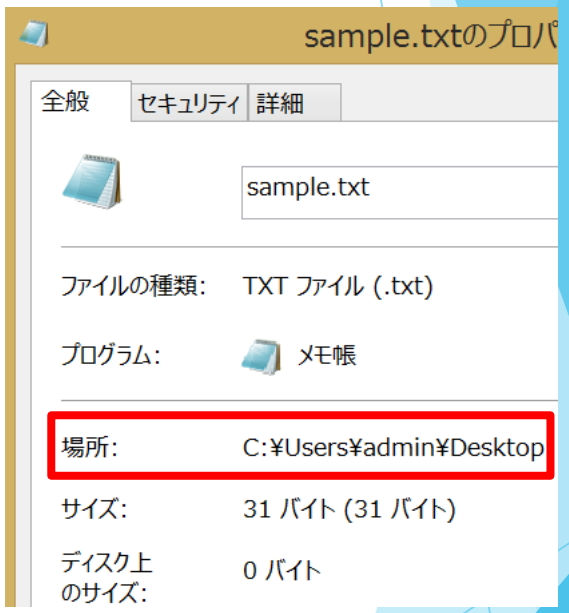
- 使用例：セーブデータ・ロード
- ゲームの状態(ステータス、マップでの位置)をファイル(セーブデータ)に書き込み保存
- ファイル(セーブデータ)を読み込み、そこに保存されているゲームの状態を使ってセーブしたときの状態を復元する

使用例②

- 使用例：ハイスコアランキング
- プレイヤーのスコアとファイルに書かれたスコアを比較し、ランキングに入っていれば書き込む
- おそらく、オンラインゲームではそのハイスコアを管理したファイルがネットワークを通じて共有されている

open()①

- 変数 = open(場所, モード)
- 場所：ファイルの場所とファイル名を指定
  使いたいファイルのプロパティから参照
- 例：C:¥Users¥admin¥Desktop
- 書き方は二通りある




```
file = open(r"C:\Users\admin\Desktop\sample.txt")
file = open("C:\\Users\\admin\\Desktop\\sample.txt")
```

open()②

- 変数 ＝ open(場所, モード)
- モード：指定したファイルをどのように使用するかを決める、無記入の場合は読み取り(”r”)

r	読み込み専用。書き込みはできない。 ファイルが存在しないときはエラーになる。
w	書き込み専用。読み込みはできない。 ファイルが存在していない場合は新規作成。
a	追加書き込み専用。読み込みはできない。 ファイルがない場合は新規作成。
r+	読み書き両用。 ファイルがない場合はエラー。
w+	読み書き両用。 ファイルがある場合は「w」と同じ処理。
a+	追記・読み書き両用。 ファイルがない場合は新規作成。

https://sites.google.com/site/kuraitlab/programing-language/python/file-operations

close()

- 変数.close()
- ファイルを開いた場合、最終的に必ず書く必要がある文
- 変数はopen(場所, モード)を入れたもの
- 開けたら閉める、の精神で

```
file = open("C:\\Users\\admin\\Desktop\\sample.txt","r")
print(file.read())
file.close()
```

with文

- with open(場所, モード) as 変数:
- open()とclose()の処理をまとめたもの
- ブロックでファイルの開閉を管理できるため閉じ忘れが無い、基本的にこちらを使う

```
file = open("C:\\Users\\admin\\Desktop\\sample.txt","r")
print(file.read())
file.close()
```

↓

```
with open (r"C:\Users\admin\Desktop\sample.txt") as file:
    print(file.read())
```

読み込み

- 変数.read()

ファイルを全て読み込み、それを一つの文字列として扱う

- 変数.readlines()

ファイルを全て読み込み、それを1行毎に文字列としてリストの要素に割り当てられる

- 変数.readline()

ファイルの1行を読み込み、それを文字列として扱う

書き込み

- 変数.write(“文字列”)

文字列を引数に取り、ファイルに書き込む

- 変数.writelines(“””文字列”””)

文字列のリストを引数に取り、ファイルに書き込む

エスケープシーケンス

- 文字の中には1文字では表せない文字も存在する(改行など)
- ¥+文字によって表現できるようになる

http://www.pythonweb.jp/tutorial/string/index2.html

エスケープシーケンス	意味
¥¥	「¥」文字そのもの
¥'	シングルクオーテーション
¥"	ダブルクオーテーション
¥a	ベル
¥b	バックスペース
¥f	改ページ
¥r	キャリッジリターン
¥n	改行
¥t	水平タブ
¥v	垂直タブ
¥N{name}	Unicode データベース中で名前 name を持つ文字
¥uxxxx	16ビットの16進数値xxxxを持つUnicode文字
¥Uxxxxxxxx	32ビットの16進数値xxxxxxxxを持つUnicode文字
¥ooo	8進数oooを持つASCII文字
¥xhh	16進数hhを持つASCII文字
¥0	NULL
¥+(改行)	文字列を途中で改行する

Unicode①

- Pythonでは文字をUnicodeという形式で表している。Unicodeにおいて
- 数字の「0」＝ 48
- アルファベットの「A」＝ 65
- ひらがなの「あ」＝12354
- というような形でコードポイントという数字がそれぞれ割り当てられている

http://docs.python.jp/2/howto/unicode.html

Unicode②

- ord(“文字”)
- 文字のコードポイントを取得

  ord(“A”) = 65
- chr(数字)
- その数字の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