



# Pythonのファイル操作

作成、編集、書き込み、読み込み



## openメソッドについて

openメソッドはファイル操作をするためのメソッド

```
f = open(file, mode='r')
```

第1引数(file): ファイル名

第2引数(mode): r 読み込み用を開く

w ファイルを作成する。同名のファイルが存在する場合上書きする。

x 新規ファイルを作成する。同名のファイルが存在する場合エラーを返す。

a 書き込みように開き、ファイルが存在する場合は中身の最後に上書きする。



## withステートメントについて

withステートメントは最後にファイルを自動的に閉じてくれる  
close()メソッドを書き忘れる心配がないため非常に便利なステートメント

使い方

```
with open('ファイル名', 'w') as f:  
    pass
```



## ファイル作成

以下のように記述する。

```
with open("ファイル名",w) as f:  
    pass
```

開いたが最後書き込まなかったら中身はすべて消える。

開く前に読み込みで開き、ファイルの中身を変数に保存しておくべき



## ファイル書き込み

- ・データを書き換える場合

```
with open(filename,'w') as f:  
    f.write('test\n')  
    print('Hello', 'World')
```

- ・データを追記する場合

```
with open(filename,'a') as f:  
    f.write('test\n')  
    print('Hello', 'World')
```



## ファイル読み込み

with open(filename, 'r') as f:

while True:

line = f.readline() #1行ずつ読み込み

print(line, end="") #end=""は改行しないための処理

if not line:

break



## ファイルの読み書き

with open(filename, 'w+') as f:

```
f.read()
```

```
f.write('test')
```

w+の場合は一番最初に読み込みを実行しても中身がない。(新しいファイルを作成している為)

with open(filename, 'r+') as f:

```
f.read()
```

```
f.write('test')
```

r+の場合は一番最初に読み込みができる。  
読み込みの場合ファイルは作成しないので、ファイルが存在しない場合はエラーになる。



## ファイル内のデータの位置移動

seekメソッドを使用して文字列を移動する

```
with open(filename, 'r') as f:
```

```
    print(f.read(1))
```

```
    f.seek(1)
```

```
    print(f.read(1))
```

```
    f.seek(2)
```

ファイルに何か入力してから実行してみてください。





## stringパッケージについて(テンプレート)

下記のコードを基準に説明する。

```
import string
```

```
s = """"\  
こんにちは $name さん  
$contents  
さようなら  
""""
```

```
t = string.Template(s)  
text = t.substitute(name='ボンクラ', contents='今日はいい天気ですね')  
print(text)
```

stringパッケージのテンプレートメソッドを使うことによって\$から始まる変数に文字を代入することができる。