

## ファイル入出力

ファイルの入出力は今まで画面に出力していた結果をファイルに出力するものです。結果を保存できるのでランキング機能やセーブ機能を実装することが出来ます。

まず初めにテキストデータのファイル入出力について説明しましょう。

### 【必須ヘッダー】

<stdio.h>

### 【基本的な関数】

ファイルの制御	fopen	fclose	
ファイルの読み込み	fgetc	fgets	fscanf
ファイルの書き込み	fputc	fputs	fprintf

---

### ● fopen

#### 【宣言】

```
FILE *fopen(const char *filename, const char *mode);
```

#### 【概略】

ファイル名 (filename) で指定したファイルをモード(mode) で開きます

#### 【引数】

filename : 開きたいファイルのファイル名

mode : オープンモード

#### 【戻り値】

そのファイルに関する情報を格納した FILE 構造体のポインタ  
ファイルが正しく開けなかった場合、NULL を返す

#### 【解説】

まず FILE 構造体というのは、ファイルを開いた時にそのファイルに関する情報を保存する構造体の事です。ファイルの入出力には基本的にこの構造体を使います。

この `fopen` という関数は第一引数にファイル名を `char` 型で指定し、第二引数でファイルの開き方を指定します。そして、ファイルが正常に開かれた場合そのファイルの `FILE` 構造体のポインタが返ってきます。

mode	意味
r	テキストファイルを読み込み用に開く
w	テキストファイルを書き込み用に開く
a	テキストファイルを追記用に開く
rb	バイナリファイルを読み込み用に開く
wb	バイナリファイルを書き込み用に開く
ab	バイナリファイルを追記用に開く
r+	テキストファイルを読み書き両用に開く
w+	テキストファイルを読み書き両用に開く
a+	テキストファイルを読み書き両用で追加あるいは作成する
rb+	バイナリファイルを読み書き両用に開く
wb+	バイナリファイルを読み書き両用に開く
ab+	バイナリファイルを読み書き両用で追加あるいは作成する

モードはたくさんありますが、基本的に「r: 読み込み (Read)」「w: 書き込み (Write)」「a: 追記 (add)」の3つからなっています。それに「+: 読み書き両用」「b: バイナリ (binary)」を組み合わせることによってモードは成り立っています。

モードに「r」が付いている場合、指定したファイルが存在しない場合、関数は失敗します (返り値 `NULL`)。一方「w」「a」が付いている場合、ファイルが存在しない場合、指定したファイル名で自動でファイルが作成されます。

「a」で開いた場合、内容を消すことなく末尾から追記することが可能ですが、「w」で開いた場合ファイルを開いた時点で内容は消えてしまいます。

- **fclose**

- 【宣言】

- ```
int fclose(FILE *fp);
```

- 【概略】

- `fopen` でオープンされたファイルポインタ (`fp`) で示されるファイルをクローズします。

- 【引数】

- `fp` : `fopen` でオープンしたファイルのファイルポインタ

- 【戻り値】

- 関数が成功した場合 0 を返し、関数が失敗した場合 EOF を返す

- 【サンプルプログラム】

|                                                               |                                                                |
|---------------------------------------------------------------|----------------------------------------------------------------|
| <pre>FILE *fp; fp = fopen("test.txt", "w"); fclose(fp);</pre> | <pre>//FILE構造体のポインタ //test.txtを書き込みで開く //ファイルポインタをクローズする</pre> |
|---------------------------------------------------------------|----------------------------------------------------------------|

---

以下サンプルプログラムの `test.txt` の内容を下記とする

|                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>The radius of 85 centimeters, it's the distance I can reach around Now I will turn and swing my hands above it, can you keep yourself away to see?</pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|

- **fgetc**

- 【宣言】

- ```
int fgetc(FILE *fp);
```

- 【概略】

- `fopen` でオープンされたファイルポインタ (`fp`) を引数にし、そこから 1 文字読み取って返します

- 【引数】

- `fp` : `fopen` でオープンしたファイルのファイルポインタ

- 【戻り値】

関数が成功した場合読み取った文字を返し、関数が失敗した場合 EOF (-1) を返す

#### 【サンプルプログラム】

```
FILE *fp; //FILE構造体のポインタ
fp = fopen("test.txt", "r"); //test.txtを読み込みで開く
char c = fgetc(fp); //test.txtから1文字取得しcに代入
fclose(fp); //ファイルポインタをクローズする
```

このプログラムの結果、変数 c に 'T' が格納されます。

#### ● fgets

##### 【宣言】

```
char *fgets(char *Buf, int n, FILE *fp);
```

##### 【概略】

ファイルポインタ (fp) で示されるファイルから Buf に 1 行、または n 文字格納します

##### 【引数】

Buf : 読み取った文字列を格納する文字列バッファ  
n : 読み取る最大文字数  
fp : fopen でオープンしたファイルのファイルポインタ

##### 【戻り値】

関数が成功した場合読み取った文字列を格納場所を返し、関数が失敗した場合 NULL を返す

##### 【解説】

ファイルを現在位置から「n-1」文字分、または改行 or 終端まで読み取り、Buf に格納します。

#### 【サンプルプログラム】

```
FILE *fp; //FILE構造体のポインタ
fp = fopen("test.txt", "r"); //test.txtを読み込みで開く
char str[64]; //格納する文字列バッファの宣言
fgets(str, 64, fp); //test.txtから1行取得しstrに格納
fclose(fp); //ファイルポインタをクローズする
```

このプログラムの結果、文字列 `str` に“The radius of 85 centimeters,”が格納されます。このプログラムでは第二引数に 64 を指定したので 64 文字格納出来ます。もしこの引数が 8 のような少ない場合“The rad”が格納されます。

- `fscanf`

**【宣言】**

```
int fscanf(FILE *fp, const char *format, ...);
```

**【概略】**

ファイルポインタ (`fp`) で指定したファイルから書式つきで格納する。

`scanf` のファイル版。

**【引数】**

`fp` : `fopen` でオープンしたファイルのファイルポインタ

`format` : 書式制御文字列

... : 可変個引数。この引数を書式指定に従い変換します  
書式指定文字列はこの引数と同数必要

**【戻り値】**

関数が成功した場合読み取った項目数を返し、関数が失敗した場合 EOF (-1) を返す

**【サンプルプログラム】**

```
FILE *fp; //FILE構造体のポインタ
fp = fopen("test.txt", "r"); //test.txtを読み込みで開く
char str[4][64]; //格納する文字列バッファの宣言
int num; //数字を格納する変数の宣言
fscanf (fp, "%s %s %s %d %s", str[0], str[1], str[2], &num, str[3]);
fclose (fp); //ファイルポインタをクローズする
```

このプログラムの結果、下記のように成

変数	str[0]	str[1]	str[2]	num	str[3]
値	The	radius	of	85	centimeters,

- **fputc**

- 【宣言】

- ```
int fputc( int c, FILE *fp );
```

- 【概略】

- 1文字ファイルに書き込み

- 【引数】

- c** : ファイルに書き出す文字

- fp** : fopen でオープンしたファイルのファイルポインタ

- 【戻り値】

- 成功すると、出力した文字を返し、失敗すると、EOF(-1)を返します。

- 【サンプルプログラム】

|                                                                                           |                                                                                                 |
|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| <pre>FILE *fp; char c = 'T'; fp = fopen("test.txt", "w"); fputc(c, fp); fclose(fp);</pre> | <pre>//FILE構造体のポインタ //text.txtに格納する文字 //test.txtを書き込みで開く //ファイルに1文字書き出す //ファイルポインタをクローズする</pre> |
|-------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|

このプログラムの結果、ファイル (test.txt) に「T」が書き出されま  
す。

- **fputs**

- 【宣言】

- ```
int fputs( const char *str, FILE *fp );
```

- 【概略】

- 文字列を、まとめてファイル書き込み

- 【引数】

- str** : ファイルに書き出す文字列

- fp** : fopen でオープンしたファイルのファイルポインタ

- 【戻り値】

- 成功すると、0以上の値を返し、失敗すると、EOF(-1)を返します。

- 【サンプルプログラム】

```

FILE *fp; //FILE構造体のポインタ
char str[] = "I love Lolita."; //test.txtに格納する文字列
fp = fopen("test.txt", "w"); //test.txtを書き込みで開く
fputs(str, fp); //ファイルに文字列を書き出す
fclose(fp); //ファイルポインタをクローズする

```

このプログラムの結果、ファイル (test.txt) に「I love Lolita.」が書き出されます。

- **fprintf**

- 【宣言】

```
int fprintf(FILE *fp, const char *format, ...);
```

- 【概略】

ファイルポインタ (fp) で指定したファイルに書式つきで書き出しする。

printf のファイル版。

- 【引数】

fp : fopen でオープンしたファイルのファイルポインタ

format : 書式制御文字列

... : 可変個引数。この引数を書式指定に従い変換します

書式指定文字列はこの引数と同数必要

- 【戻り値】

負の値 : エラー

それ以外 : 書き込んだ文字数

- 【サンプルプログラム】

```

FILE *fp; //FILE構造体のポインタ
fp = fopen("test.txt", "w"); //test.txtを書き込みで開く
char str[2][64] = {"I have", "wife"}; //書き出す文字列の宣言
int num = 3; //書き出す値の変数の宣言
fprintf(fp, "%s %d %s\n", str[0], num, str[1]); //ファイルに書き込み
fclose(fp); //ファイルポインタをクローズ

```

このプログラムの結果によりファイル (test.txt) に「I have 3 wife」が出力されます。