第６回Ｃ言語講座

１．　ファイルに保存する

今までは出力の対象はコンソール画面でした。しかし、コンソールでは表示しきれないデータや残しておきたいデータなどはファイルに保存しなければなりません。

まずはファイルへの保存方法をやります。

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/*
C言語講座サンプル6
ファイルに保存する
by y.t
*/
#include <stdio.h> /*入出力のヘッダーファイル*/
/*メイン関数*/
int main()
{
	FILE *fp;				//ファイル構造体へのポインタ
	char fname[256];
	char str[1024];

	printf("ファイル名を入力してください。¥n");
	scanf("%s", fname);
	fp = fopen(fname, "w");		//ファイルを書き込みモードで開く
	printf("文章を入れてください¥n");
	scanf("%s", str);
	fprintf(fp, "%s", str);		//ファイルに文字を出力
	fclose(fp);					//ファイルを閉じる
	return 0;
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

ファイルはさまざまなモードで開かれます。保存では書き込みモードを使います。そのほかのモードは表1．1を参照してください。

表1．1ファイルのオープンモード

引数の文字	モードの意味
w	書き込みモードで空のファイルを開く。
r	読み込みモードで既存のファイルを開く。
a	追加書き込みモードでファイルを開く。
w+	読み込みと書き込みの両方のモードで空のファイルを開く。
r+	読み込みと書き込みの両方のモードで既存のファイルを開く。
a+	読み込みと追加の両方のモードでファイルを開く。

※　引数文字の前に”b”をつけるとバイナリモードでファイルを開きます。

ファイルを扱うには FILE 構造体を用います。これにはファイルに関する情報が格納されます。次に fopen 関数を使用します。fopen 関数にはファイル名とファイルのオープンモードを設定します。fopen 関数はファイル構造体のポインタを返す関数なので FILE 構造体はポインタ変数で宣言します。

ファイルにデータを書き込むには fprintf 関数を使います。printf と違い、書き込み先のファイルを指定して書き込むことができます。書き込み先は第一引数に FILE 構造体へのポインタを指定することで行います。

ファイルの最後は fclose 関数を使ってファイルを閉じます。これを行わないと書き込んだ内容がファイルに保存されません。

２．　ファイルからの読み込み

ファイルの書き込みの次はファイルの読み込みです。

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
/*
C言語講座サンプル2
ファイルの読み込み
by y.t
*/
#include <stdio.h> /*入出力のヘッダーファイル*/
/*メイン関数*/
int main()
{
        FILE *fp;                       //ファイル構造体へのポインタ
        char fname[256];
        char str[1024];
        printf("ファイル名を入力してください。¥n");
        scanf("%s", fname);
        fp = fopen(fname, "r");         //ファイルを読み込みモードで開く
        if (fp == NULL)  printf("ファイルが見つかりません¥n");
        fscanf(fp, "%s", str);
        printf("%s¥n", str);
        fclose(fp);                                     //ファイルを閉じる
```

```
	return 0;
}
```

//

ファイルを開く段階は前回と同じです。しかし、今回は読み込むのでファイルは読み込みモードで開きます。次に、ファイルが存在しない場合 fopen 関数は NULL を返すので、もし NULL が返されたときはファイルが存在しないことを示す必要があります。

ファイルを読むには fscanf 関数を使います。この関数の第一引数に FILE 構造体のポインタを指定すると、そのファイルからデータを読み込みます。使い方自体は scanf 関数と変わりません。

３．　型のキャスト

変数には決められたデータ型が存在します。しかし、変数を一時的に別なデータ型を変換したいときがしばしばあります。そのような時は型のキャストを使ってデータ型を変換してやります。

//

```
/*
C言語講座サンプル3
型のキャスト
by y.t
*/
#include <stdio.h> /*入出力のヘッダーファイル*/
/*メイン関数*/
int main()
{
	float f;
	printf("実数を入力¥n");
	scanf("%f", &f);
	printf("f = %f¥n", f);
	printf("(int) = %d¥n", (int)f);		//ここでキャスト
	return 0;
}
```

//

キャストしたいデータ型を括弧でくくって変数の前に記述すればキャストは完了します。キャストは一時的なものであり、キャストによってもとのデータ型が変更されるということはありません。

４．　本日の演習

１．　ファイルに任意の文字を書き込む

２．　ファイルから文字列を読み込む。

５．　本日の関数

表5．1．本日の関数

関数	引数	戻り値	処理内容
FILE *fopen(const char *filename, const char mode);	filename ファイル名 mode アクセス許可の種類	開いているファイルへのポインタ	ファイルを開く。 必須ヘッダー stdio.h
FILE *fprintf(FILE *stream, const char *format [, argument]…);	stream FILE 構造体へのポインタ format 書式指定文字列 argument 省略可能な引数	出力したバイト数	ファイルへの書き込み 必須ヘッダー stdio.h
int scanf(FILE *stream, const char *format [, argument]…);	stream FILE 構造体へのポインタ format 書式指定文字列 argument 省略可能な引数	代入された変数の数	ファイルからの読み込み
int fclose(FILE *stream);	stream FILE 構造体へのポインタ	正常にファイルを閉じた=0 異常終了=EOF	ファイルを閉じる 必須ヘッダー stdio.h