

# プログラミング基礎

2023年度1Q 火曜日 3, 4 時限(10:45~12:25)  
金曜日 1~ 4 時限( 8:50~12:25)

工学院 情報通信系

中山実, 渡辺義浩

伊藤泉, 小杉哲

TA: 小泊大輝, 千脇彰悟

5/26(金) 8:50~12:25

- 第13回「入出力関数」

1. いろいろな入出力関数

2. ファイル入出力

# 出力関数 printf, 入力関数 scanf

```
#include <stdio.h>
```

```
int main(void)
{
```

```
    char char1;
    int intnum;
    float floatnum;
```

```
    printf("Input character:¥n");
    scanf("%c", &char1);
```

```
    printf("Input int num:¥n");
    scanf("%d", &intnum);
```

```
    printf("Input float num:¥n");
    scanf("%f", &floatnum);
```

```
    printf("character: %c, Int num %d, Float num %4.2f¥n", char1, intnum,
floatnum);
```

```
    return 0;
```

```
}
```

## 実行結果

Input character:

a

Input int num:

5

Input float num:

3.14

character: a, Int num 5, Float num 3.14

# 出力関数 printf, 入力関数 scanf

```
#include <stdio.h>

int main(void)
{
    char str[100];
    printf("Input words. ¥n");
    scanf("%s", str);
    printf("The words are: %s ¥n", str);

    return 0;
}
```

## 実行結果 1

Input words.  
This is a pen.  
The words are: This

## 実行結果 2

Input words.  
ThisIsAPen  
The words are: ThisIsAPen

scanf関数で読むと空白が読めない

# 1 行入力 gets, 1行出力 puts

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    char str[100];
```

```
    printf("Input words. ¥n");
```

```
    gets(str);
```

```
    printf("The words are: ¥n");
```

```
    puts(str);
```

```
    return 0;
```

```
}
```

- char \*gets(char \*buffer);
  - 空白も読める
  - 宣言した配列のサイズ以上のデータでも読み込んでしまう
  - 危険なため, 基本使わない

## 実行結果

```
Input words.
```

```
This is a pen.
```

```
The words are:
```

```
This is a pen.
```

# 1 行入力 gets, 1行出力 puts

```
#include <stdio.h>
```

```
int main(void)
{
    char str[100];
    printf("Input words. ¥n");
    gets(str);
    printf("The words are: ¥n");
    puts(str);

    return 0;
}
```

- int puts(const char \*string);
  - 書式指定なしで 1 行出力ができる
  - 改行も自動的

## 実行結果

```
Input words.
This is a pen.
The words are:
This is a pen.
```

# 1 文字入力 getchar, 1文字出力 putchar

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int ch;
```

```
    printf("Input words. ¥n");
```

```
    while ((ch = getchar()) != EOF)
```

```
    {
```

```
        putchar(ch);
```

```
        fflush(stdout);
```

```
    }
```

```
    return 0;
```

```
}
```

- int getchar(void);

- 1文字入力

- int putchar(int c);

- 1文字出力

# 1文字入力 getchar, 1文字出力 putchar

```
#include <stdio.h>

int main(void)
{
    int ch;

    printf("Input words. \n");
    while ((ch = getchar()) != EOF)
    {
        putchar(ch);
        fflush(stdout);
    }

    return 0;
}
```

- EOF(end of file)が入力されるまで、getcharとputcharを繰り返す
- EOF
  - “ctrl+z” (Windows)
  - “ctrl+d” (Mac, Unix)



# 1文字入力 getchar, 1文字出力 putchar

```
#include <stdio.h>

int main(void)
{
    int ch;

    printf("Input words. \n");
    while ((ch = getchar()) != EOF)
    {
        putchar(ch);
        fflush(stdout);
    }

    return 0;
}
```

## 実行結果例

```
Input words.
This is a pen.
This is a pen.
I have a pen.
I have a pen.
“ctrl + {d or z}”
```

\*終わらない場合は、一度コンソール以外のウィンドウを選んでみる。

# 5/26(金) 8:50~12:25

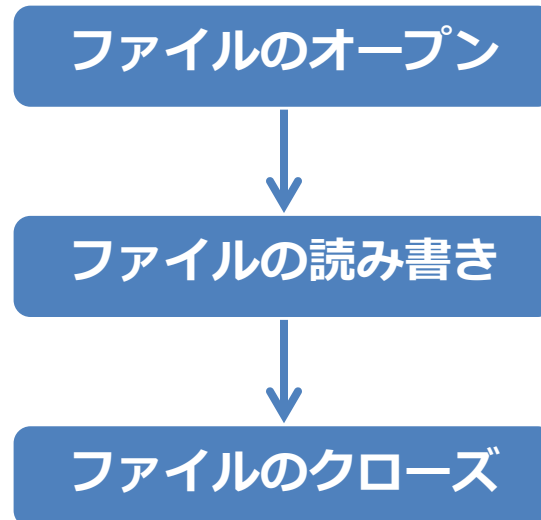
- 第13回「入出力関数」

1. いろいろな入出力関数

2. ファイル入出力

# ファイルの入出力

- データの入出力
  - 画面やキーボードを用いた入出力
  - ファイルを用いた入出力



# ファイルの入出力

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char filename[] = "sample1.txt";

    fp = fopen(filename, "w");
    if (fp == NULL)
    {
        printf("File was not opened. \n");
        return 1;
    }
    else
    {
        printf("File was opened. \n");
    }
    fclose(fp);
    printf("File is closed. \n");
    return 0;
}
```

- ファイルポインタ  
– ファイル情報の格納先

# ファイルの入出力

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char filename[] = "sample1.txt";

    fp = fopen(filename, "w");
    if (fp == NULL)
    {
        printf("File was not opened. ¥n");
        return 1;
    }
    else
    {
        printf("File was opened. ¥n");
    }
    fclose(fp);
    printf("File is closed. ¥n");
    return 0;
}
```

- ファイル名変数の定義と初期化

# ファイルの入出力

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char filename[] = "sample1.txt";

    fp = fopen(filename, "w");
    if (fp == NULL)
    {
        printf("File was not opened. ¥n");
        return 1;
    }
    else
    {
        printf("File was opened. ¥n");
    }
    fclose(fp);
    printf("File is closed. ¥n");
    return 0;
}
```

- ファイルのオープン

# ファイルの入出力

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

↑  
ファイル名

↑  
モード

モード	操作	説明
r	読み込み	指定ファイルがあれば開いて読み込み。 なければNULLを返す。
w	書き込み	指定ファイルがあれば上書き。 なければ新規作成。
a	追加書き込み	指定ファイルがあれば、ファイルの末尾に追加。なければ新規作成。

# ファイルの入出力

```
#include <stdio.h>
```

```
int main(void)
{
```

```
    FILE *fp;
```

```
    char filename[] = "sample1.txt";
```

```
    fp = fopen(filename, "w");
```

```
    if (fp == NULL)
```

```
    {
```

```
        printf("File was not opened. \n");
```

```
        return 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("File was opened. \n");
```

```
    }
```

```
    fclose(fp);
```

```
    printf("File is closed. \n");
```

```
    return 0;
```

```
}
```

- エラー処理



# ファイルの入出力

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char filename[] = "sample1.txt";

    fp = fopen(filename, "w");
    if (fp == NULL)
    {
        printf("File was not opened. ¥n");
        return 1;
    }
    else
    {
        printf("File was opened. ¥n");
    }
    fclose(fp);
    printf("File is closed. ¥n");
    return 0;
}
```

- ファイルのクローズ

# ファイルの入出力

```
#include <stdio.h>

int main(void)
{
    FILE *fp;
    char filename[] = "sample1.txt";

    fp = fopen(filename, "w");
    if (fp == NULL)
    {
        printf("File was not opened. \n");
        return 1;
    }
    else
    {
        printf("File was opened. \n");
    }
    fclose(fp);
    printf("File is closed. \n");
    return 0;
}
```

## 実行結果

sample1.txtが新しく作成  
される

```
#include <stdio.h>
```

```
int main(void)
{
```

```
    FILE *fp;
```

```
    char filename[] = "sample1.txt";
```

```
    fp = fopen(filename, "w");
```

```
    if (fp == NULL)
```

```
    {
```

```
        printf("File was not opened. %n");
```

```
        return 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("File was opened. %n");
```

```
    }
```

```
    fputs("Beef%n", fp);
```

```
    fputs("Pork%n", fp);
```

```
    fputs("Chicken%n", fp);
```

```
    fclose(fp);
```

```
    printf("File was closed. %n");
```

```
    return 0;
```

```
}
```

## 実行結果

"sample1.txt"生成  
以下ファイルの中身

Beef

Pork

Chicken

- int fputs(const char \*string, FILE \*stream);
  - ファイル(fp)に文字列を書き込み

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    FILE *fp;
```

```
    char filename[] = "sample2.txt";
```

```
    int i;
```

```
    int data[6] = {34, 45, 56, 67, 78, 22};
```

```
    fp = fopen(filename, "w");
```

```
    if (fp == NULL)
```

```
    {
```

```
        printf("File was not opened ¥n");
```

```
        return 1;
```

```
    }
```

```
    else
```

```
    {
```

```
        printf("File was opened. ¥n");
```

```
    }
```

```
    for (i = 0; i < 6; i++)
```

```
        fprintf(fp, "%d¥n", data[i]);
```

```
    fclose(fp);
```

```
    printf("File was closed. ¥n");
```

```
    return 0;
```

```
}
```

## 実行結果

"sample2.txt"生成  
以下ファイルの中身

34

45

56

67

78

22

- fprintf(FILE \*fp, 書式, 変数)
  - ファイルにデータを書き込み

```
#include <stdio.h>
#define NUM 20
```

```
int main(void)
{
    FILE *fp;
    char str1[NUM], str2[NUM], str3[NUM];
    char filename[] = "sample1.txt";

    fp = fopen(filename, "r");
    if (fp == NULL)
    {
        printf("File is not open. ¥n");
        return 1;
    }
    else
    {
        printf("File is open. ¥n");
    }
    fgets(str1, NUM, fp);
    fgets(str2, NUM, fp);
    fgets(str3, NUM, fp);

    printf("%s", str1);
    printf("%s", str2);
    printf("%s", str3);

    fclose(fp);
    printf("File is closed. ¥n");
    return 0;
}
```

"sample1.txt"

Beef Pork Chicken
-------------------------

- ファイルは同じプロジェクトフォルダ直下に置く

```

#include <stdio.h>
#define NUM 20

int main(void)
{
    FILE *fp;
    char str1[NUM], str2[NUM], str3[NUM];
    char filename[] = "sample1.txt";

    fp = fopen(filename, "r");
    if (fp == NULL)
    {
        printf("File is not open. ¥n");
        return 1;
    }
    else
    {
        printf("File is open. ¥n");
    }
    fgets(str1, NUM, fp);
    fgets(str2, NUM, fp);
    fgets(str3, NUM, fp);

    printf("%s", str1);
    printf("%s", str2);
    printf("%s", str3);

    fclose(fp);
    printf("File is closed. ¥n");
    return 0;
}

```

- char \*fgets(char \*str, int n, FILE \*fp);
  - 現在の stream 位置から最初の改行文字 (¥n) まで (改行文字を含める)
  - またはストリームの終わりまで
  - あるいは読み込まれた文字数が n-1 と同じになるまで
  - 以上のうち、最も早いもののまでの文字を読み取る

```

#include <stdio.h>
#define NUM 20

int main(void)
{
    FILE *fp;
    char str1[NUM], str2[NUM], str3[NUM];
    char filename[] = "sample1.txt";

    fp = fopen(filename, "r");
    if (fp == NULL)
    {
        printf("File is not open. ¥n");
        return 1;
    }
    else
    {
        printf("File is open. ¥n");
    }
    fgets(str1, NUM, fp);
    fgets(str2, NUM, fp);
    fgets(str3, NUM, fp);

    printf("%s", str1);
    printf("%s", str2);
    printf("%s", str3);

    fclose(fp);
    printf("File is closed. ¥n");
    return 0;
}

```

## 実行結果

```

File is open.
Beef
Pork
Chicken
File is closed.

```

```
#include <stdio.h>
```

```
int main(void) {  
    FILE *fp;  
    char filename[] = "sample2.txt";  
    int i;  
    int data[20];  
  
    fp = fopen(filename, "r");  
  
    if(fp == NULL){  
        printf("File is not open. \n");  
        return 1;  
    }  
    else printf("File is open. \n");  
    for (i=0; i<6; i++){  
        fscanf(fp, "%d", &data[i]);  
        printf("%d,%d\n",data[i],2*data[i]);  
    }  
    fclose(fp);  
    printf("File is closed. \n");  
  
    return 0;  
}
```

"sample2.txt"

34  
45  
56  
67  
78  
22



```
#include <stdio.h>
```

```
int main(void) {  
    FILE *fp;  
    char filename[] = "sample2.txt";  
    int i;  
    int data[20];  
  
    fp = fopen(filename, "r");  
  
    if(fp == NULL){  
        printf("File is not open. ¥n");  
        return 1;  
    }  
    else printf("File is open. ¥n");  
    for (i=0; i<6; i++){  
        fscanf(fp, "%d", &data[i]);  
        printf("%d,%d¥n", data[i], 2*data[i]);  
    }  
    fclose(fp);  
    printf("File is closed. ¥n");  
  
    return 0;  
}
```

- fscanf(FILE \*fp, 書式, 読込変数)
  - ファイルからデータを読み込む
  - 戻り値は, 読取に成功した入力項目の個数

```
#include <stdio.h>
```

```
int main(void) {  
    FILE *fp;  
    char filename[] = "sample2.txt";  
    int i;  
    int data[20];  
  
    fp = fopen(filename, "r");  
  
    if(fp == NULL){  
        printf("File is not open. ¥n");  
        return 1;  
    }  
    else printf("File is open. ¥n");  
    for (i=0; i<6; i++){  
        fscanf(fp, "%d", &data[i]);  
        printf("%d,%d¥n",data[i],2*data[i]);  
    }  
    fclose(fp);  
    printf("File is closed. ¥n");  
  
    return 0;  
}
```

## 実行結果

```
File is open.  
34,68  
45,90  
56,112  
67,134  
78,156  
22,44  
File is closed.
```