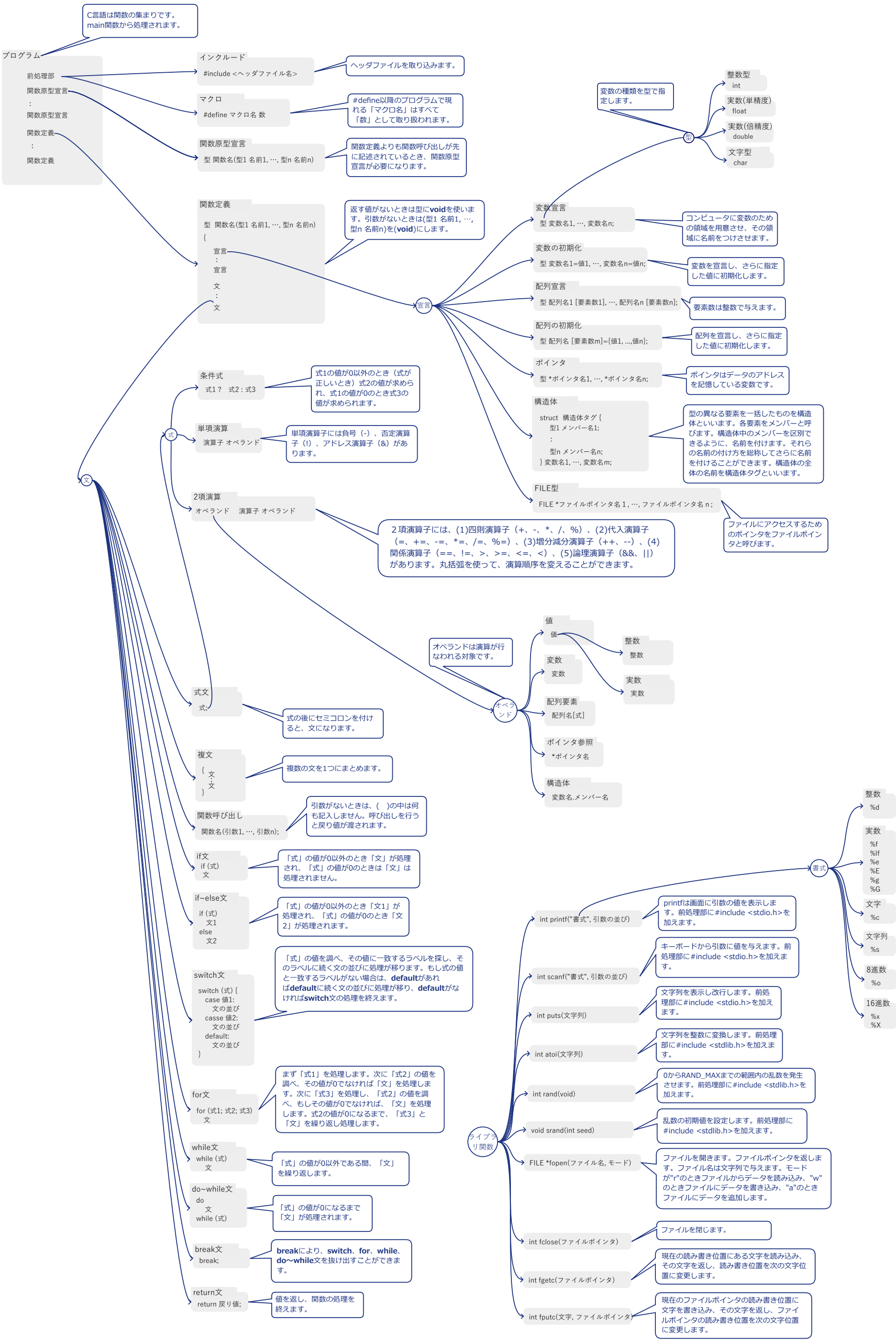
C言語リファレンス


C言語は関数の集まりです。main関数から処理されます。
プログラム
前処理部
関数原型宣言
：
関数原型宣言
関数定義
：
関数定義
インクルード
#include <ヘッダファイル名>
ヘッダファイルを取り込みます。
マクロ
#define マクロ名 数
#define以降のプログラムで現れる「マクロ名」はすべて「数」として取り扱われます。
関数原型宣言
型 関数名(型1 名前1, …, 型n 名前n)
関数定義よりも関数呼び出しが先に記述されているとき、関数原型宣言が必要になります。
関数定義
型 関数名(型1 名前1, …, 型n 名前n)
{
宣言
：
宣言
文
：
文
返す値がないときは型にvoidを使います。引数がないときは(型1 名前1, …, 型n 名前n)を(void)にします。
宣言
変数の種類を型で指定します。
型
整数型
int
実数(単精度)
float
実数(倍精度)
double
文字型
char
変数宣言
型 変数名1, …, 変数名n;
コンピュータに変数のための領域を用意させ、その領域に名前をつけさせます。
変数の初期化
型 変数名1=値1, …, 変数名n=値n;
変数を宣言し、さらに指定した値に初期化します。
配列宣言
型 配列名1 [要素数1], …, 配列名n [要素数n];
要素数は整数で与えます。
配列の初期化
型 配列名 [要素数m]={値1, ...,値n};
配列を宣言し、さらに指定した値に初期化します。
ポインタ
型 *ポインタ名1, …, *ポインタ名n;
ポインタはデータのアドレスを記憶している変数です。
構造体
struct 構造体タグ {
型1 メンバー名1;
：
型n メンバー名n;
} 変数名1, …, 変数名m;
型の異なる要素を一括したものを構造体といいます。各要素をメンバーと呼びます。構造体中のメンバーを区別できるように、名前を付けます。それらの名前の付け方を総称してさらに名前を付けることができます。構造体の全体の名前を構造体タグといいます。
FILE型
FILE *ファイルポインタ名１, …, ファイルポインタ名ｎ;
ファイルにアクセスするためのポインタをファイルポインタと呼びます。
文
式
条件式
式1？ 式2 : 式3
式1の値が0以外のとき（式が正しいとき）式2の値が求められ、式1の値が0のとき式3の値が求められます。
単項演算
演算子 オペランド
単項演算子には負号（-）、否定演算子（!）、アドレス演算子（&）があります。
2項演算
オペランド 演算子 オペランド
２項演算子には、(1)四則演算子（+、-、*、/、%）、(2)代入演算子（=、+=、-=、*=、/=、%=）、(3)増分減分演算子（++、--）、(4)関係演算子（==、!=、>、>=、<=、<）、(5)論理演算子（&&、||）があります。丸括弧を使って、演算順序を変えることができます。
オペランドは演算が行なわれる対象です。
オペランド
値
値
整数
整数
実数
実数
変数
変数
配列要素
配列名[式]
ポインタ参照
*ポインタ名
構造体
変数名.メンバー名
式文
式;
式の後にセミコロンを付けると、文になります。
複文
{
文
：
文
}
複数の文を1つにまとめます。
関数呼び出し
関数名(引数1, …, 引数n);
引数がないときは、( )の中は何も記入しません。呼び出しを行うと戻り値が渡されます。
if文
if (式)
文
「式」の値が0以外のとき「文」が処理され、「式」の値が0のときは「文」は処理されません。
if~else文
if (式)
文1
else
文2
「式」の値が0以外のとき「文1」が処理され、「式」の値が0のとき「文2」が処理されます。
switch文
switch (式) {
case 値1:
文の並び
casse 値2:
文の並び
default:
文の並び
}
「式」の値を調べ、その値に一致するラベルを探し、そのラベルに続く文の並びに処理が移ります。もし式の値と一致するラベルがない場合は、defaultがあればdefaultに続く文の並びに処理が移り、defaultがなければswitch文の処理を終えます。
for文
for (式1; 式2; 式3)
文
まず「式1」を処理します。次に「式2」の値を調べ、その値が0でなければ「文」を処理します。次に「式3」を処理し、「式2」の値を調べ、もしその値が0でなければ、「文」を処理します。式2の値が0になるまで、「式3」と「文」を繰り返し処理します。
while文
while (式)
文
「式」の値が0以外である間、「文」を繰り返します。
do~while文
do
文
while (式)
「式」の値が0になるまで「文」が処理されます。
break文
break;
breakにより、switch、for、while、do～whileを抜け出すことができます。
return文
return 戻り値;
値を返し、関数の処理を終えます。
書式
整数
%d
実数
%f
%lf
%e
%E
%g
%G
文字
%c
文字列
%s
8進数
%o
16進数
%x
%X
ライブラリ関数
int printf("書式", 引数の並び)
printfは画面に引数の値を表示します。前処理部に#include <stdio.h>を加えます。
int scanf("書式", 引数の並び)
キーボードから引数に値を与えます。前処理部に#include <stdio.h>を加えます。
int puts(文字列)
文字列を表示し改行します。前処理部に#include <stdio.h>を加えます。
int atoi(文字列)
文字列を整数に変換します。前処理部に#include <stdlib.h>を加えます。
int rand(void)
0からRAND_MAXまでの範囲内の乱数を発生させます。前処理部に#include <stdlib.h>を加えます。
void srand(int seed)
乱数の初期値を設定します。前処理部に#include <stdlib.h>を加えます。
FILE *fopen(ファイル名, モード)
ファイルを開きます。ファイルポインタを返します。ファイル名は文字列で与えます。モードが"r"のときファイルからデータを読み込み、"w"のときファイルにデータを書き込み、"a"のときファイルにデータを追加します。
int fclose(ファイルポインタ)
ファイルを閉じます。
int fgetc(ファイルポインタ)
現在の読み書き位置にある文字を読み込み、その文字を返し、読み書き位置を次の文字位置に変更します。
int fputc(文字, ファイルポインタ)
現在のファイルポインタの読み書き位置に文字を書き込み、その文字を返し、ファイルポインタの読み書き位置を次の文字位置に変更します。