# 第5章 制御構造（条件分岐）

## 対応C規格

* **主要対象:** C90
* **学習内容:** if文、else文、switch文、条件演算子による分岐処理

## 学習目標

この章を完了すると、以下のことができるようになります：

* if文を使った条件分岐ができる
* else if文で複数条件を処理できる
* switch文による多分岐処理ができる
* 条件演算子を適切に使える
* ネストした条件分岐を理解する

## 概要と詳細

### 条件分岐の世界へようこそ

私たちは日常生活で常に「もし〜なら」という判断をしています。「もし雨が降ったら傘を持つ」「もし電車が遅れたらバスで行く」など、状況に応じて行動を変えています。

プログラミングでも同じように、条件によって処理を変える必要があります。この章では、C言語で条件分岐を実現する方法を学びます。

### なぜ条件分岐が必要なのか？

条件分岐がないプログラムは、いつも同じ処理しかできません。それでは： - ユーザーの入力に応じた処理ができない - エラーが発生しても対処できない - 状況に応じた最適な処理を選べない

条件分岐を使うことで、プログラムに「知能」を与えることができます。

### if文の基本（もし〜なら）

if文は最も基本的な条件分岐です。日本語の「もし〜なら〜する」をそのままプログラムで表現できます。

#### 基本構文

if (条件式) {  
 /\* 条件が真（正しい）の場合に実行される処理 \*/  
}

**重要なポイント**： - 条件式は括弧 () で囲む - 処理部分は波括弧 {} で囲む（1行でも推奨） - 条件が偽（正しくない）の場合は、if文の中は実行されない

#### 実生活に例えた単純なif文

#include <stdio.h>  
  
int main(void)  
{  
 int rain\_probability = 70; /\* 降水確率70% \*/  
   
 if (rain\_probability >= 50) {  
 printf("傘を持っていきましょう\n");  
 }  
   
 int temperature = 35; /\* 気温35度 \*/  
   
 if (temperature >= 30) {  
 printf("熱中症に注意！水分補給を忘れずに\n");  
 }  
   
 return 0;  
}

#### 波括弧の重要性

/\* 危険な書き方（波括弧なし） \*/  
if (score >= 80)  
 printf("合格です\n");  
 printf("おめでとう！\n"); /\* これは常に実行される！ \*/  
  
/\* 安全な書き方（波括弧あり） \*/  
if (score >= 80) {  
 printf("合格です\n");  
 printf("おめでとう！\n"); /\* 両方とも条件が真の時だけ実行 \*/  
}

### if-else文（そうでなければ）

現実の判断では「もし〜なら〜する、そうでなければ〜する」という二者択一の場面がよくあります。if-else文はまさにこの状況を表現します。

#### 基本構文

if (条件式) {  
 /\* 条件が真（正しい）の場合の処理 \*/  
} else {  
 /\* 条件が偽（正しくない）の場合の処理 \*/  
}

**ポイント**： - どちらか一方の処理が必ず実行される - elseは「それ以外すべて」を意味する

#### 実用的なif-else文の例

/\* 成人判定 \*/  
int age = 19;  
  
if (age >= 20) {  
 printf("成人です\n");  
 printf("選挙権があります\n");  
} else {  
 printf("未成年です\n");  
 printf("あと%d年で成人です\n", 20 - age);  
}  
  
/\* パスワード認証 \*/  
int password = 1234;  
int input = 1235;  
  
if (password == input) {  
 printf("ログイン成功！\n");  
 printf("ようこそ！\n");  
} else {  
 printf("パスワードが違います\n");  
 printf("もう一度お試しください\n");  
}  
  
/\* 営業判定 \*/  
int hour = 18;  
  
if (hour >= 9 && hour < 17) {  
 printf("営業中です\n");  
} else {  
 printf("営業時間外です\n");  
}

#### よくある使い方：エラー処理

int balance = 1000; /\* 残高 \*/  
int withdraw = 1500; /\* 引き出し額 \*/  
  
if (balance >= withdraw) {  
 balance -= withdraw;  
 printf("引き出し成功\n");  
 printf("残高: %d円\n", balance);  
} else {  
 printf("残高不足です\n");  
 printf("現在の残高: %d円\n", balance);  
}

### if-else if文（複数の選択肢）

3つ以上の選択肢から選ぶ場面では、if-else if文を使います。上から順番に条件をチェックし、最初に真になった処理を実行します。

#### 基本構文

if (条件1) {  
 /\* 条件1が真の場合 \*/  
} else if (条件2) {  
 /\* 条件1は偽で、条件2が真の場合 \*/  
} else if (条件3) {  
 /\* 条件1,2は偽で、条件3が真の場合 \*/  
} else {  
 /\* すべての条件が偽の場合 \*/  
}

**重要な特徴**： - 上から順番にチェックされる - 最初に真になった部分だけ実行される - それ以降の条件はチェックされない

#### 成績判定システムの例

int score = 85;  
  
printf("点数: %d点\n", score);  
  
if (score >= 90) {  
 printf("成績: A（優秀）\n");  
 printf("素晴らしい成績です！\n");  
} else if (score >= 80) {  
 printf("成績: B（良好）\n");  
 printf("よく頑張りました！\n");  
} else if (score >= 70) {  
 printf("成績: C（普通）\n");  
 printf("合格です\n");  
} else if (score >= 60) {  
 printf("成績: D（可）\n");  
 printf("ギリギリ合格です\n");  
} else {  
 printf("成績: F（不可）\n");  
 printf("もう少し頑張りましょう\n");  
}

#### 時間帯による挨拶の例

int hour = 14; /\* 14時（午後2時） \*/  
  
if (hour >= 5 && hour < 12) {  
 printf("おはようございます\n");  
} else if (hour >= 12 && hour < 17) {  
 printf("こんにちは\n");  
} else if (hour >= 17 && hour < 21) {  
 printf("こんばんは\n");  
} else {  
 printf("夜遅いですね、お疲れさまです\n");  
}

#### 順序が重要な理由

/\* 間違った順序 \*/  
int age = 15;  
  
if (age >= 0) {  
 printf("生まれています\n"); /\* 15歳でもここで止まる！ \*/  
} else if (age >= 13) {  
 printf("中学生以上\n"); /\* 実行されない \*/  
} else if (age >= 18) {  
 printf("成人\n"); /\* 実行されない \*/  
}  
  
/\* 正しい順序（厳しい条件から） \*/  
if (age >= 18) {  
 printf("成人\n");  
} else if (age >= 13) {  
 printf("中学生以上\n");  
} else if (age >= 0) {  
 printf("生まれています\n");  
}

### ネストしたif文（入れ子構造）

if文の中にさらにif文を書くことで、より複雑な条件判断ができます。

/\* 天気と気温による服装アドバイス \*/  
int temperature = 25;  
int is\_raining = 0; /\* 0:晴れ、1:雨 \*/  
  
if (temperature >= 25) {  
 printf("暖かい日です\n");  
 if (is\_raining) {  
 printf("半袖で傘を持っていきましょう\n");  
 } else {  
 printf("半袖で大丈夫です\n");  
 }  
} else if (temperature >= 15) {  
 printf("過ごしやすい気温です\n");  
 if (is\_raining) {  
 printf("長袖で傘を持っていきましょう\n");  
 } else {  
 printf("長袖がおすすめです\n");  
 }  
} else {  
 printf("寒い日です\n");  
 if (is\_raining) {  
 printf("コートと傘が必要です\n");  
 } else {  
 printf("コートを着ていきましょう\n");  
 }  
}

#### ネストを避ける工夫

ネストが深くなると読みにくくなるので、論理演算子を使って簡潔に書くこともできます：

/\* ネストを使った場合 \*/  
if (age >= 18) {  
 if (has\_license) {  
 printf("運転できます\n");  
 }  
}  
  
/\* 論理演算子を使った場合 \*/  
if (age >= 18 && has\_license) {  
 printf("運転できます\n");  
}

### 条件式の詳細

#### 比較演算子

int a = 10, b = 5;  
  
if (a == b) printf("等しい\n");   
if (a != b) printf("等しくない\n");   
if (a < b) printf("a は b より小さい\n");   
if (a <= b) printf("a は b 以下\n");   
if (a > b) printf("a は b より大きい\n");   
if (a >= b) printf("a は b 以上\n");

#### 論理演算子の組み合わせ

int age = 25;  
int income = 3000000;  
  
/\* AND演算子 \*/  
if (age >= 20 && income >= 2000000) {  
 printf("ローン審査に通りました\n");  
}  
  
/\* OR演算子 \*/  
if (age < 18 || age > 65) {  
 printf("特別料金が適用されます\n");  
}  
  
/\* NOT演算子 \*/  
if (!(age >= 18)) {  
 printf("18歳未満です\n");  
}

### switch文（多分岐の効率的な処理）

#### switch文って何？

プログラムを書いていると、「この変数の値が1なら処理A、2なら処理B、3なら処理C…」というように、一つの変数の値によって多くの分岐をしたい場面があります。

これをif-else ifで書くと：

if (choice == 1) {  
 printf("処理A\n");  
} else if (choice == 2) {  
 printf("処理B\n");  
} else if (choice == 3) {  
 printf("処理C\n");  
} else if (choice == 4) {  
 printf("処理D\n");  
} else {  
 printf("その他の処理\n");  
}

このように長くなってしまいます。switch文を使うと、もっとスッキリ書けます！

#### 日常生活での例え

switch文は「自動販売機」のようなものです：

1. ボタンを押す（変数の値）
2. 押したボタンに応じて違う商品が出る（caseごとの処理）
3. 存在しないボタンを押したら何も出ない（default処理）

int button = 2; /\* 2番のボタンを押した \*/  
  
switch (button) {  
 case 1:  
 printf("コーラが出ました\n");  
 break;  
 case 2:  
 printf("オレンジジュースが出ました\n");  
 break;  
 case 3:  
 printf("お茶が出ました\n");  
 break;  
 default:  
 printf("そのボタンには商品がありません\n");  
 break;  
}

#### 基本構文

switch (変数または式) {  
 case 値1:  
 /\* 値1の場合の処理 \*/  
 break;  
 case 値2:  
 /\* 値2の場合の処理 \*/  
 break;  
 case 値3:  
 /\* 値3の場合の処理 \*/  
 break;  
 default:  
 /\* どの値にも一致しない場合の処理 \*/  
 break;  
}

#### 曜日判定の例

int day = 3;  
  
switch (day) {  
 case 1:  
 printf("月曜日\n");  
 break;  
 case 2:  
 printf("火曜日\n");  
 break;  
 case 3:  
 printf("水曜日\n");  
 break;  
 case 4:  
 printf("木曜日\n");  
 break;  
 case 5:  
 printf("金曜日\n");  
 break;  
 case 6:  
 printf("土曜日\n");  
 break;  
 case 7:  
 printf("日曜日\n");  
 break;  
 default:  
 printf("無効な曜日です\n");  
 break;  
}

#### break文の重要性

break文を忘れると、次のcaseも実行されます（フォールスルー）：

int grade = 'B';  
  
switch (grade) {  
 case 'A':  
 printf("優秀\n");  
 /\* break がないので次のcaseも実行される \*/  
 case 'B':  
 printf("良好\n");  
 /\* break がないので次のcaseも実行される \*/  
 case 'C':  
 printf("普通\n");  
 break;  
 default:  
 printf("要努力\n");  
 break;  
}  
/\* grade が 'B' の場合、"良好" と "普通" の両方が出力される \*/

#### 意図的なフォールスルー

時には意図的にbreakを省略することもあります：

char ch = 'a';  
  
switch (ch) {  
 case 'a':  
 case 'e':  
 case 'i':  
 case 'o':  
 case 'u':  
 printf("母音です\n");  
 break;  
 default:  
 printf("子音です\n");  
 break;  
}

### 条件演算子（三項演算子）の復習

条件分岐の簡潔な書き方として条件演算子があります。

int a = 10, b = 5;  
int max;  
  
/\* if-else文での記述 \*/  
if (a > b) {  
 max = a;  
} else {  
 max = b;  
}  
  
/\* 条件演算子での記述 \*/  
max = (a > b) ? a : b;  
  
printf("最大値: %d\n", max);

### 条件式での注意点

#### 代入と比較の混同

int x = 5;  
  
/\* NG: 代入になってしまう \*/  
if (x = 10) {  
 printf("常に実行される\n"); /\* x に 10 が代入され、常に真 \*/  
}  
  
/\* OK: 比較演算子を使用 \*/  
if (x == 10) {  
 printf("x が 10 の場合のみ実行\n");  
}

#### 浮動小数点数の比較

double d = 0.1 + 0.2;  
  
/\* NG: 浮動小数点の誤差で期待通りにならない可能性 \*/  
if (d == 0.3) {  
 printf("等しい\n");  
}  
  
/\* OK: 誤差を考慮した比較 \*/  
if (d >= 0.29999 && d <= 0.30001) {  
 printf("ほぼ等しい\n");  
}

## サンプルコード

### if文の基本使用例

プログラムファイル: examples/if\_basic.c

基本的なif文、if-else文の使用方法を学習します。

### 複雑な条件分岐

プログラムファイル: examples/complex\_conditions.c

複数条件を組み合わせた分岐処理を学習します。

### switch文の実例

プログラムファイル: examples/switch\_demo.c

switch文を使った多分岐処理を学習します。

### コンパイルと実行

# examples ディレクトリに移動  
cd examples  
  
# C90準拠でコンパイル  
gcc -std=c90 -Wall -Wextra -pedantic if\_basic.c -o if\_basic  
  
# 実行  
./if\_basic

## 演習課題

### 基礎問題

1. **年齢判定プログラム**
   * 年齢を入力して、幼児・児童・学生・成人・高齢者に分類するプログラムを作成してください
2. **電卓プログラム**
   * 2つの数値と演算子（+, -, \*, /）を入力して計算結果を表示するプログラムをswitch文で作成してください
3. **成績判定システム**
   * 点数を入力してA〜Fの成績を判定するプログラムを作成してください

### 応用問題

1. **うるう年判定**
   * 年を入力してうるう年かどうかを判定するプログラムを作成してください
   * 条件: 4で割り切れる、ただし100で割り切れる年は平年、400で割り切れる年はうるう年
2. **BMI計算と判定**
   * 身長と体重を入力してBMIを計算し、肥満度を判定するプログラムを作成してください
3. **三角形の判定**
   * 3つの辺の長さを入力して、三角形として成立するか、どの種類の三角形かを判定してください

## コンパイル方法

この章では以下のMakefileを使用してコンパイルができます：

# 全ての例題をコンパイル  
make all  
  
# 特定のプログラムをコンパイル  
make if\_basic  
  
# 実行  
make run  
  
# クリーンアップ  
make clean

## 規格による違い

### C90での制限事項

* switch文の case ラベルは整数定数のみ
* 複合文内での変数宣言は先頭のみ

### C99以降の拡張

* switch文でlong long型が使用可能
* ブロック内での変数宣言位置の制限緩和

## よくある間違い

### 1. セミコロンの位置

/\* NG: if文の後にセミコロン \*/  
if (condition);  
{  
 printf("常に実行される\n"); /\* 条件に関係なく実行 \*/  
}  
  
/\* OK: 正しい記述 \*/  
if (condition) {  
 printf("条件が真の場合のみ実行\n");  
}

### 2. switch文でのbreak忘れ

/\* NG: break忘れでフォールスルー \*/  
switch (value) {  
 case 1:  
 printf("1です\n");  
 /\* break; を忘れると次のcaseも実行される \*/  
 case 2:  
 printf("2です\n");  
 break;  
}  
  
/\* OK: 適切なbreak文 \*/  
switch (value) {  
 case 1:  
 printf("1です\n");  
 break;  
 case 2:  
 printf("2です\n");  
 break;  
}

### 3. 複雑な条件式での括弧不足

/\* NG: 意図しない優先順位 \*/  
if (a == 1 || b == 2 && c == 3) {  
 /\* && が || より優先される \*/  
}  
  
/\* OK: 括弧で明示 \*/  
if ((a == 1) || (b == 2 && c == 3)) {  
 /\* 意図が明確 \*/  
}

## 次の章へ

条件分岐を理解したら、[制御構造（ループ）](../control-loop/README.md) に進んでください。

## 参考資料

* [if文詳細](https://ja.cppreference.com/w/c/language/if)
* [switch文詳細](https://ja.cppreference.com/w/c/language/switch)
* [条件演算子](https://ja.cppreference.com/w/c/language/operautor_other)

# サンプルコード

## conditional\_operator.c

```c / 条件演算子（三項演算子）の使い方（C90準拠） \* \* このプログラムは、条件演算子を使った \* 簡潔な条件分岐の書き方を示します。 \* 規格: C90準拠 \*/ #include <stdio.h>

int main(void) { /\* 変数の宣言（C90では先頭で宣言） */ int a, b; int max, min; int age; int score; int bonus; int number; int abs\_value; char* result; char \*status;

/\* 2つの数の最大値・最小値を求める \*/  
printf("2つの整数を入力してください: ");  
scanf("%d %d", &a, &b);  
  
/\* 条件演算子を使った最大値・最小値の判定 \*/  
max = (a > b) ? a : b;  
min = (a < b) ? a : b;  
  
printf("最大値: %d\n", max);  
printf("最小値: %d\n", min);  
  
/\* 年齢による判定 \*/  
printf("\n年齢を入力してください: ");  
scanf("%d", &age);  
  
status = (age >= 20) ? "成人" : "未成年";  
printf("あなたは%sです。\n", status);  
  
/\* 成績による合否判定 \*/  
printf("\n点数を入力してください (0-100): ");  
scanf("%d", &score);  
  
result = (score >= 60) ? "合格" : "不合格";  
printf("判定: %s\n", result);  
  
/\* ネストした条件演算子（読みやすさには注意） \*/  
printf("成績評価: ");  
printf("%c\n", (score >= 90) ? 'A' :  
 (score >= 80) ? 'B' :  
 (score >= 70) ? 'C' :  
 (score >= 60) ? 'D' : 'F');  
  
/\* 条件演算子と通常の計算を組み合わせる \*/  
bonus = (score >= 80) ? 10 : 0;  
printf("ボーナス点: %d\n", bonus);  
printf("最終得点: %d\n", score + bonus);  
  
/\* 絶対値の計算 \*/  
printf("\n整数を入力してください: ");  
scanf("%d", &number);  
  
abs\_value = (number >= 0) ? number : -number;  
printf("%d の絶対値は %d です。\n", number, abs\_value);  
  
return 0;

}```

## if\_basic.c

```c / if文の基本的な使い方（C90準拠） \* \* このプログラムは、if文、if-else文、if-else if文の \* 基本的な使い方を示します。 \* 規格: C90準拠 \*/ #include <stdio.h>

int main(void) { /\* 変数の宣言（C90では先頭で宣言） \*/ int score;

printf("点数を入力してください (0-100): ");  
scanf("%d", &score);  
  
/\* 単純なif文 \*/  
if (score == 100) {  
 printf("満点です！素晴らしい！\n");  
}  
  
/\* if-else文 \*/  
if (score >= 60) {  
 printf("合格です\n");  
} else {  
 printf("不合格です\n");  
}  
  
/\* if-else if文による成績判定 \*/  
printf("成績評価: ");  
if (score >= 90) {  
 printf("A (優秀)\n");  
} else if (score >= 80) {  
 printf("B (良好)\n");  
} else if (score >= 70) {  
 printf("C (普通)\n");  
} else if (score >= 60) {  
 printf("D (可)\n");  
} else {  
 printf("F (不可)\n");  
}  
  
/\* ネストしたif文 \*/  
if (score >= 60) {  
 printf("おめでとうございます！");  
 if (score >= 80) {  
 printf("優秀な成績です！");  
 }  
 printf("\n");  
}  
  
return 0;

}```

## switch\_basic.c

```c / switch文の基本的な使い方（C90準拠） \* \* このプログラムは、switch文による多分岐処理の \* 基本的な使い方を示します。 \* 規格: C90準拠 \*/ #include <stdio.h>

int main(void) { /\* 変数の宣言（C90では先頭で宣言） \*/ int menu\_choice; char grade; int month;

/\* メニュー選択の例 \*/  
printf("=== レストランメニュー ===\n");  
printf("1. ハンバーガー (500円)\n");  
printf("2. ピザ (800円)\n");  
printf("3. パスタ (700円)\n");  
printf("4. サラダ (400円)\n");  
printf("選択してください (1-4): ");  
scanf("%d", &menu\_choice);  
  
switch (menu\_choice) {  
 case 1:  
 printf("ハンバーガーを注文しました。500円です。\n");  
 break;  
 case 2:  
 printf("ピザを注文しました。800円です。\n");  
 break;  
 case 3:  
 printf("パスタを注文しました。700円です。\n");  
 break;  
 case 4:  
 printf("サラダを注文しました。400円です。\n");  
 break;  
 default:  
 printf("無効な選択です。\n");  
 break;  
}  
  
/\* 成績評価の例 \*/  
printf("\n成績を入力してください (A-F): ");  
scanf(" %c", &grade); /\* 空白文字を読み飛ばすため、%cの前に空白を入れる \*/  
  
switch (grade) {  
 case 'A':  
 case 'a':  
 printf("優秀: 90点以上\n");  
 break;  
 case 'B':  
 case 'b':  
 printf("良好: 80-89点\n");  
 break;  
 case 'C':  
 case 'c':  
 printf("普通: 70-79点\n");  
 break;  
 case 'D':  
 case 'd':  
 printf("可: 60-69点\n");  
 break;  
 case 'F':  
 case 'f':  
 printf("不可: 60点未満\n");  
 break;  
 default:  
 printf("無効な成績です。\n");  
 break;  
}  
  
/\* 月から季節を判定する例（フォールスルーの活用） \*/  
printf("\n月を入力してください (1-12): ");  
scanf("%d", &month);  
  
printf("季節: ");  
switch (month) {  
 case 12:  
 case 1:  
 case 2:  
 printf("冬\n");  
 break;  
 case 3:  
 case 4:  
 case 5:  
 printf("春\n");  
 break;  
 case 6:  
 case 7:  
 case 8:  
 printf("夏\n");  
 break;  
 case 9:  
 case 10:  
 case 11:  
 printf("秋\n");  
 break;  
 default:  
 printf("無効な月です\n");  
 break;  
}  
  
return 0;

}```