

プログラミング基礎演習A



if文による条件分岐
比較演算子, 論理演算子

if文

```
if(条件) {
```

条件が満たされるとき実行される命令

```
}
```

Tabキーを1回押してインデントをつける

文の最後でも波括弧}の後にセミコロン;はつけない

◆ 条件が満たされるときだけ 内が実行される

プログラム例

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

```
int main()
```

```
{  
    int score;
```

```
    printf("得点を入力してください:¥n");  
    scanf("%d", &score);
```

```
    if(score >= 90) {  
        printf("90点以上です¥nよくできました¥n");  
    }
```

```
得点を入力してください:  
93  
90点以上です  
よくできました
```



93
score

if-else文

```
if(条件) {
```

条件が満たされるとき実行される命令 ①

```
} else {
```

条件が満たされるとき実行される命令 ②

```
}
```

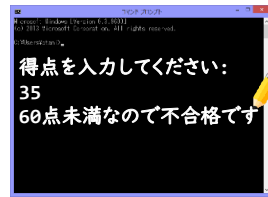
◆ 条件が満たされるとき①, 条件が満たされるとき②が実行される

プログラム例

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

```
int main()
{
    int score;

    printf("得点を入力してください:¥n");
    scanf("%d", &score);
    if(score < 60) {
        printf("60点未満なので不合格です¥n");
    } else {
        printf("60点以上なので合格です¥n");
    }
}
```



35
score

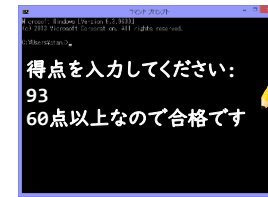
5

プログラム例

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

```
int main()
{
    int score;

    printf("得点を入力してください:¥n");
    scanf("%d", &score);
    if(score < 60) {
        printf("60点未満なので不合格です¥n");
    } else {
        printf("60点以上なので合格です¥n");
    }
}
```



93
score

6

インデント幅のルール

- ◆ 最初のインデント幅(Tabの数)は0とする
- ◆ {を1つ書くたびにインデント幅を1増やす
- ◆ }を書く前にインデント幅を1減らす

最初
は0

```
int main()
{
    int score;
    printf("得点を入力してください:¥n");
    scanf("%d", &score);
    if(score < 60) {
        printf("60点未満なので不合格です¥n");
    } else {
        printf("60点以上なので合格です¥n");
    }
}
```

+1

-1

7

複数の条件分岐

```
if(条件1) {
    条件1が満たされるとき実行される命令
} else if(条件2) {
    条件1が満たされず条件2が満たされるとき実行される命令
} else if(条件3) {
    条件1,2が満たされず条件3が満たされるとき実行される命令
} else {
    条件1,2,3が満たされないとき実行される命令
}
```

else {...}
の部分は
なくてもよい

さらに
条件を
追加しても
よい

8

プログラム例

```
if(score < 60) {  
    printf("60点未満なので「不可」です\n");  
} else if(score < 70) {  
    printf("60点以上70点未満なので「可」です\n");  
} else if(score < 80) {  
    printf("70点以上80点未満なので「良」です\n");  
} else if(score < 90) {  
    printf("80点以上90点未満なので「優」です\n");  
} else {  
    printf("90点以上なので「秀」です\n");  
}
```

9

比較演算子

- ◆ 式1 == 式2 「式1は式2と等しい」
- ◆ 式1 != 式2 「式1は式2と等しくない」
- ◆ 式1 > 式2 「式1は式2より大きい」
- ◆ 式1 < 式2 「式1は式2より小さい」
- ◆ 式1 >= 式2 「式1は式2より大きいか等しい」
- ◆ 式1 <= 式2 「式1は式2より小さいか等しい」



ノットイコール,
大なりイコール,
小なりイコール
と音で覚えよう



1 < x <= 3のように、比較演算子を
2つ以上並べることはできない

10

論理演算子

- ◆ (条件1) || (条件2) 「条件1または条件2」
- ◆ (条件1) && (条件2) 「条件1かつ条件2」
- ◆ !(条件) 「条件でない」

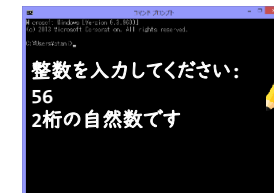
例)

(1 < x) && (x <= 3) : 1 < x ≤ 3
(x == 3) || (x == 5) : xは3か5

11

プログラム例

```
#include <stdio.h>  
  
int main()  
{  
    int x;  
  
    printf("整数を入力してください:\n");  
    scanf("%d", &x);  
    if((10 <= x) && (x <= 99)) {  
        printf("2桁の自然数です\n");  
    } else {  
        printf("2桁の自然数ではありません\n");  
    }  
}
```



56
x

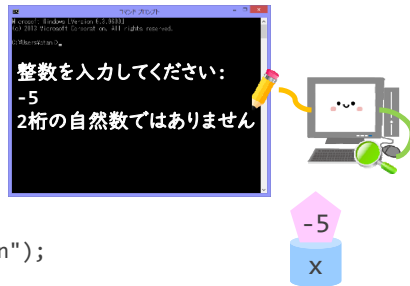
12

プログラム例

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

```
int main()
{
    int x;

    printf("整数を入力してください:¥n");
    scanf("%d", &x);
    if((10 <= x) && (x <= 99)) {
        printf("2桁の自然数です¥n");
    } else {
        printf("2桁の自然数ではありません¥n");
    }
}
```



13

プログラムにおける条件判定

発展

- ◆ 条件の部分が0か否かで真偽を判定

- ◆ 0なら「条件は満たされない」
- ◆ 0以外なら「条件は満たされる」

- ◆ 比較演算子や論理演算子を含む式は、式が成り立たないときに0, 成り立つときに0以外となる。

```
if(point < 60) {
    ↓ pointが70のとき0, 50のとき0以外の値になる
}
```

- ◆ 条件に値を書いてもよい。

```
if(0) {
    if(1) {
```

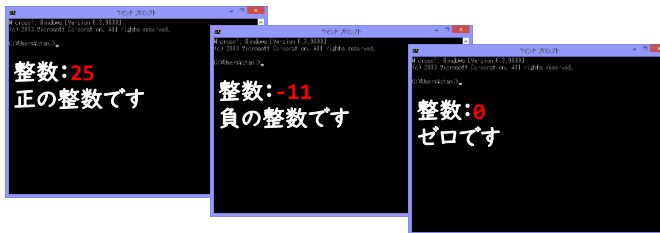
14

演習

整数を入力すると、負の整数、正の整数、ゼロのいずれであるかが表示されるプログラムを作成せよ。

プログラム名はe4とすること。

<実行例>



赤字は実行時にキーボードから入力する部分

15

演習 ～手順～

1. どんなプログラムにも絶対にあるのは?

- メイン関数
- まずは中身が空のメイン関数を書く

2. 必要なヘッダファイルは?

- 入力や出力にはstdio.hが必要
- #includeの行を書く

3. 必要な変数は?

- 入力する整数を代入する変数
- 型と変数名を決めて変数を宣言する

ここまでは
スラスラ
できるよね?



```
#include <stdio.h>

int main()
{
    int x;
}
```

16

演習 ～手順～

4. 処理の順番は？

- 「整数:」を表示する (`printf`)
- 変数に値を読み込む (`scanf`)
- x が0より小さかったら (`if`文始まり)
 - 「負の整数です」と表示する (`printf`)
- そうではなくて x が0より大きかったら (`else if`)
 - 「正の整数です」と表示する (`printf`)
- そうではなかったら(`else`)
 - 「ゼロです」と表示する (`printf`)

