

第二回C言語講習

条件分岐～変数のスコープ

目次

- 条件分岐
 - if
 - if-else
 - 条件式
- 制御構文
 - while,do-while
 - for
 - break,continue
 - switch
- 変数のスコープ
 - グローバル変数

if文

もし～ならば

```
if(条件式A){  
    Aが真だったらする処理  
}
```

if~else文

```
if(条件式A){  
    Aが偽だったときの処理  
}else{  
    Aが真だったときの処理  
}
```

```
if(条件式){  
    ほげ  
}else if(条件式){  
    ふが  
}else {  
    ぴよ  
}
```



```
if(条件式A){  
    if(条件式B){  
        AかつBが真のときの処理  
    }else{  
        Aが真でBが偽のときの処理  
    }  
    Aが真のときの処理  
}
```

条件式

真と偽

C言語における真と偽

真(true)とは0以外の値

偽(false)とは0

比較演算子

比較演算子

演算子	数学における記号	使用例
<code>==</code>	$=$	$a = b$
<code>!=</code>	\neq	$a \neq b$
<code><</code>	$<$	$a < b$
<code><=</code>	\leq	$a \leq b$
<code>></code>	$>$	$a > b$
<code>>=</code>	\geq	$a \geq b$

注意点

$$0 < x < 100$$

みたいな書き方はできない

論理演算子

論理演算子

演算子	数学の記号	使用例	備考
&&	\wedge (AND)	<code>0<x&& x<100</code>	
	\vee (OR)	<code>a!=0 b!=0</code>	
!	\neg (NOT)	<code>!(a==0&&b==0)</code>	真偽を反転する

目次

- 条件分岐
 - if
 - if-else
 - 条件式
- 制御構文 @
 - while,do-while
 - for
 - break,continue
 - switch
- 変数のスコープ
 - グローバル変数

制御構文

繰り返し処理

ループ

while文

～の間


```
while(条件式A){  
    条件式Aが真の間繰り返す処理B  
}
```

注意点

条件式が真のままだと
いつまで
経っても終わらない
(無限ループ)

do~while文

```
do{  
    繰り返す処理  
}while(条件式)
```

最低一回は繰り返し処理を行
う

for文

whileよりも高機能


```
for(初期設定;繰返し条件;継続処理){  
    繰り返す処理  
}
```

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

for文の例(出力)

0

1

2

:

7

8

9

for文の例(処理の流れ)

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1 2

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1 2 3

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1 2 3 ...

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1 2 3 ... 8

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1 2 3 ... 8 9

for文の例

```
int i;  
for(i=0;i<10;i++){  
    printf(“%d\n”,i);  
}
```

0 1 2 3 ... 8 9

forから抜ける(この時iの値は10)

break文

ループから抜ける

今いる一番小さいループから
抜け出す

break文の例

```
while(条件式1){
```

```
    処理
```

```
    :
```

```
    if(条件式2){ //これが真だとwhileを抜ける
```

```
        break;
```

```
    }
```

```
    :
```

```
    処理
```

```
}
```

ループの外

**エラーが起きたとき
便利**

break文の例2

```
int i,j;
for(i=0;i<10;i++){
    for(j=0;j<10;j++){
        if(i==j){
            break;
        }
    }
    //breakするところに抜ける。
}
```

continue文

**break文は
ループのうしろにぬける**

**continue文は
ループの最初に戻る**

switch文

複雑な分岐に向いている

switch文

```
switch(値X){  
    case A:X==Aなら行う処理;break;  
    case B:X==Bなら行う処理;break;  
    case C:X==Cなら行う処理;break;  
    case D:X==Dなら行う処理;break;  
    :  
    default:どれでもなかったら行う処理;  
}
```


switch文

```
switch(値X){
```

```
    case A:X==Aなら行う処理;break;
```

```
    case B:X==Bなら行う処理;break;
```

```
    case C:X==Cなら行う処理;break;
```

```
    case D:X==Dなら行う処理;break;
```

```
        :
```

```
    default:どれもなかったら行う処理;
```

```
}
```


//X,A,B,C,D...は全て整数値でないといけない

**最後のbreakを忘れると
その下のすべてのcaseが実
行されてしまう。**

switch文の例

```
int x=5;
switch(x){
    case 1:printf("A");break;
    case 2:printf("B");break;
    case 3:printf("C");break;
    case 4:printf("D");break;
    case 5:printf("E");break;
    default:printf("none_case");break;
}
```

目次

- 条件分岐
 - if
 - if-else
 - 条件式
- 制御構文
 - while,do-while
 - for
 - break,continue
 - switch
- 変数のスコープ 
 - グローバル変数

変数のスコープ

**whileやforの
{ }の中でも変数宣言ができる**

{ }の中で変数宣言

```
int a;  
while(hoge){  
    int b,c;  
    a=10;  
    b=20;  
    c=a+b;  
    printf(“%d\n”,c);  
}
```

この{}の中で
宣言された変数は
{}の外では使えない

b,cの有効範囲

```
int a;
```

```
while(hoge){
```

```
    int b,c;
```

```
    a=10;
```

```
    b=20;
```

```
    c=a+b;
```

```
    printf(“%d\n”,c);
```

```
}
```

/*ここでb,cは使えない。変数がここに載っているもの意外宣言されていないと仮定 */

このような
変数の有効範囲のことをス
コープ
と言います

スコープの効果

スコープの例

```
int main(){  
    int a=10;  
    int i;  
    for(i=0;i<1;i++){  
        int a=20;  
        printf("forの中のa:%d\n",a);  
    }  
    printf("forの外のa:%d\n",a);  
}
```

スコープの例(結果)

```
[hoge@sol]$ gcc scopetest.c -o scopetest
```

```
[hoge@sol]$ ./scopetest
```

```
forの中のa20
```

```
forの外のア10
```

スコープの例

```
int main(){  
    int a=10;  
    int i;  
    for(i=0;i<1;i++){  
        int a=20;  
        printf("forの中のa:%d\n",a);  
    }  
    printf("forの外のa:%d\n",a);  
}
```

同じ名前の変数があった場合
より狭いスコープが優先され
る

グローバル変数

実はmain関数の外にも
変数が宣言できる

グローバル変数

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

```
int g;           //どこからでも使える
```

```
int main(){
```

```
    g=100;
```

```
    printf(“%d\n”,g);
```

```
    while(g){g--;} 
```

```
    return 0;
```

```
}
```

プログラム中のどこからでも
使える変数

どこからでも変更できる

大きなプログラムを書くときあ
まり安全じゃない

ちなみにグローバル変数じゃないものをローカル変数とか
局所変数と呼びます

まとめ

まとめ

- 条件分岐
 - if
 - if-else
 - 条件式
- 制御構文
 - while,do-while
 - for
 - break,continue
 - switch
- 変数のスコープ
 - グローバル変数

では各自演習
に入ってください