

C言語基礎

1. 変数宣言 (int & double)
2. 入出力 (printf & scanf)
3. 条件分岐 (if)
4. 繰り返し (for & while)

1.0 変数宣言：整数型と実数型

C言語では使用する変数を事前に宣言しなければならない

整数型

`int`

(倍精度)実数型

`double`

例

```
#include <stdio.h>
int main (){

    int a;
    double x,y;
    a=-2;
    x=1.2;
    y=1.0;
    return 0;
}
```

実数の場合、
少数点以下0でも
“.0”をつける

1.1 変数宣言：混合演算

四則演算に関して

イ. (整数)&(整数) \Rightarrow (整数)

※少数点以下,きり捨て

ロ. (実数)&(整数) \Rightarrow (実数)

ハ. (実数)&(実数) \Rightarrow (実数)

例

```
#include <stdio.h>
int main (){
```

```
    int a;
```

```
    double x,y,z;
```

① a=4/3; \leftarrow a=1

① x=4/3; \leftarrow x=1.0000000

② y=4.0/3 \leftarrow y=1.3333333

③ z=4.0/3.0; \leftarrow z=1.3333333

```
    return 0;
```

```
}
```

2. 入出力

printf : 端末に文字を出力 scanf : 端末から数字を入力

例 : 整数型

```
#include <stdio.h>
int main (){
    int a,b;
    scanf("%d",&a);
    b=2*a;
    printf("2倍すると%d\n",b);
    return 0;
}
```

“&”は呪文

intの場合%d

“\n”は改行

例 : 実数型

```
#include <stdio.h>
int main (){
    double a,b;
    scanf("%lf",&a);
    b=2*a;
    printf("2倍すると%f\n",b);
    return 0;
}
```

doubleの場合%lfと%f

3. 条件分岐

if文

if(条件1){

条件1を満たした場合,ここが実行される

}

else if(条件2){

条件1を満たさず,かつ条件2を満たす場合,
ここが実行される

}

else{

条件1 & 条件2の両方を満たさない場合,
ここが実行される

}

論理記号

意味		記号	意味		記号
\geq	▶	<code>>=</code>	かつ	▶	<code>&&</code>
\leq	▶	<code><=</code>	または	▶	<code> </code>
<code>=</code>	▶	<code>==</code>			
<code>≠</code>	▶	<code>!=</code>			

例

```
#include <stdio.h>
int main (){
    int a;
    scanf("%d",&a);
    if(a>=100){
        printf("100以上\n");
    }
    else{
        printf("100未満\n");
    }
    return 0;
}
```

4. 繰り返し：for文

for文

```
for(式A; 条件 ; 式B){
```

“式A”を初期条件にして、
“条件”を満たす間、“{ }”の中をループする。
ただし、ループするごとに“式B”を実行する。

```
}
```

while文

```
while(条件){
```

“条件”を満たす間、“{ }”の中をループする。

```
}
```

例：for文

```
#include <stdio.h>
int main (){
    int i;
    for(i=1;i<=10;i=i+1){
        printf("%d\n",i);
    }
    return 0;
}
```

現在のiの値に+1したものを、新しくiの値とする

例：while文

```
#include <stdio.h>
int main (){
    int i, s;
    i=1;
    s=0;
    while(i<=10){
        s=s+i;
        i=i+1;
    }
    printf("1~10までの和=%d\n",s);
    return 0;
}
```

現在のsの値に+iしたものを、新しくsの値とする