

C언어 반복문

반복문

C언어 반복문에서 for 반복문은 초기값, 조건식, 증감문으로 구성됩니다. While 반복문은 for 반복문에서 초기값과 증감문이 밖으로 나온 형태와 비슷합니다. Do while 반복문은 코드 블록을 1회 강제 실행할 때 사용합니다

C언어에서 반복문이 필요한 이유

C언어에서 0부터 시작해서 100까지의 숫자를 출력하려면 printf()로 작성된 101줄의 코드가 필요합니다.

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

    printf("%d \n", 0);
    printf("%d \n", 1);

    //...중략

    printf("%d \n", 99);
    printf("%d \n", 100);
}
```

반복문 사용

```
0
1
2
...중략
99
100
```

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

    for (i = 0; i <= 100; i++)
    {
        printf("%d \n", i);
    }
}
```

C언어 반복문의 종류

C언어 반복문의 종류에는 for문, while문, do while문이 있습니다.
반복문은 서로 전환이 가능합니다.

기호	형태	의미
<code>for</code>	<code>for (초깃값; 조건식; 증감문)</code>	조건식이 거짓이 될 때 까지 반복
<code>while</code>	<code>while (조건식)</code>	조건식이 거짓이 될 때 까지 반복
<code>do while</code>	<code>do {실행문} while (조건식)</code>	실행을 선 1회 하고, 조건식 검사로 반복 체크

For 반복문

C언의 for반복문은 선언한 변수에 초깃값을 대입해서 조건식이 거짓이 나올 때 까지 코드블록 {}사이의 실행코드를 반복합니다
반복이 가능한 이유는 조건식을 검사한 다음에 $i++$ 또는 $i--$ 등의 증감식에 의해서 변수가 자동으로 변하기때문입니다

For 반복문

숫자 0부터 9까지를 한줄씩 나열하고 싶은 경우 먼저 변수를 적절히 선언하고, for반복문의 괄호안에 초기값은 $i=0$; 조건식은 $i<10$; 그리고 증감 연산자는 $i++$ 로 지정하면 됩니다. 증감 연산자 뒤에는 ;이 없음을 유의합니다.

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

```
void main()
```

```
{
```

```
    변수 = i;
```

```
    for (i = 초기값; i < 조건식; i++ 증감문)
```

```
    {
```

```
        실행 코드;
```

```
    }
```

```
}
```

```
#include
```

```
void main()
```

```
{
```

```
    int i; //정수형 자료형(int) 조건으로 변수 i를 생성합니다. 변수값은 비어있습니다.
```

```
    for (i = 0; i < 10; i++)
```

```
    {
```

```
        printf("%d \n", i);
```

```
    }
```

```
}
```

0

1

2

3

4

5

6

7

8

9

For 반복문

For 반복문은 초기값과 증감문을 괄호 밖으로 뺄 수도 있습니다. 이 경우 괄호 안의 세미 콜론은 남겨두어야 합니다. 그리고 증감 연산자의 경우 밖으로 나올 때 세미 콜론이 붙어야 합니다. 사실 코드를 줄이는 목적이 아니라면 밖으로 뺀 구조가 좀 더 직관적으로 이해할 수 있습니다.

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

    i = 0; //초기값을 밖으로 뺍니다.

    for ( ; i < 10 ; )
    {
        printf("%d \n", i);
        i++; //증감 연산자는 조건식 밖으로 나올 때 세미콜론을 붙여줘야 합니다.
    }
}
```

초깃값과 증가문 이외에 조건식마저 for()사이에서 제거하면 for(;;) 형태의 세미콜론만 남습니다. 상태를 무한 루프라고하며 조건식이 항상 참이됩니다 코드블록사이의 반복문이 강제 중지를 하기전 까지 무한 반복됩니다. 종료하려면 visual studio를 기준으로 단축키 ctrl+c를 누르거나 실행상자를 마우스로 x클릭해야합니다

```
#include<stdio.h>
void main()
{
    int i;
    i = 0;

    for ( ; ; )
    {
        printf("%d \n", i);
        i++;
    }
}
```

0
1
2
...중략
888
889
....무한 증가

While 반복문

C언어 while반복문은 for반복문에서 초깃값과 증감문을 밖으로 뺀 구조와 거의 유사합니다. 차이점은 while()괄호 사이에 세미콜론이 없다는 점입니다.

```
#include<stdio.h>
void main()
{
    변수 = i;
    i = 초깃값;

    while (i < 조건식)
    {
        실행 코드;
        i++; // 증감 연산자
    }
}
```

초깃값 0 부터 시작해서 조건식인 i가 10 보다 클 때 까지 i++로 1씩 변수가 증가하면서 화면에 출력됩니다.

```
#include<stdio.h>
void main()
{
    int i;
    i = 0;

    while (i < 10)
    {
        printf("%d \n", i);
        i++;
    }
}
```

0
1
2
3
4
5
6
7
8
9

While 반복문에서 while을 for로 교체하고 조건식 앞 뒤에 세미콜론을 추가하면 for반복문으로 전환할수있습니다

```
#include<stdio.h>
void main()
{
    int i;
    i = 0; //초깃값

    for (; i < 10 ; )
    {
        printf("%d \n", i);
        i++; //증감문
    }
}
```

초깃값($i = 0$)과 증감문($i++$)을 조건식($i < 10$) 앞 뒤에 삽입하면, 한 줄로 표현되는 전통적인 for 반복문의 구조와 같습니다. 증감문은 괄호 안에 삽입시 뒤쪽 세미콜론 `;`이 사라짐을 유의합니다.

While 반복문은 조건식에 1을 삽입하면, for반복문처럼 무한 루프를 만들수있습니다. 1은 참이므로 while(무한 참)을 의미합니다

```
#include<stdio.h>
void main()
{
    int i;
    i = 0;

    while ( 1 )
    {
        printf("%d \n", i);
        i++;
    }
}
```

```
0
1
2
...중략
888
889
....무한 증가
```

Do while 반복문

C언어 애 while 반복문은 while반복문과 구성은 동일합니다 차이점은 코드 블록 {}사이의 반복 실행될 코드가 while(조건식)보다 앞에 위치합니다

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

    do
    {
        반복할 코드;
    }
    while ( 조건식 );
}
```

반복할 코드 블록이 do와 함께 앞으로 나오는 대신 while (조건식); 뒤에는 세미콜론이 붙어야 합니다.

i는 10이지만 i를 인쇄하라는 명령어가 while(i<10)이라는 조건식보다 앞에 나와있습니다.따라서 조건식과 무관하게 i=10이 1회는 출력됩니다 do while 반복문은 초깃값이 조건식을 만족하지않아도 1회는 강제 실행됩니다

```
#include<stdio.h>
void main()
{
    int i;
    i = 10;

    do
    {
        printf("%d \n", i);
        i++;
    }
    while (i < 10);
}
```

10

같은 조건이지만 while반복문으로 변경되면, 초깃값이 조건식을 만족하지 않을 경우 아무런 결과가 반환되지 않습니다.

```
#include<stdio.h>
void main()
{
    int i;
    i = 10;

    while (i < 10)
    {
        printf("%d \n", i);
        i++;
    }
}
```

가 없음

조건식의 참/거짓과 무관하게 강제 1회 반복을 하려면 do while을 사용합니다 반드시 조건식의 필터링을 거쳐야 한다면 while 또는 for반복문을 사용합니다