

변수의 스코프

학습내용

- 지역변수
- 전역변수

학습목표

- 지역변수의 유효범위를 알고 사용할 수 있다.
- 전역변수의 유효범위를 알고 사용할 수 있다.

지역변수

1 개요

01

변수 스코프(Scope) : 변수 참조가 가능한 유효범위

02

함수 내에 선언된 변수, Local Variable

03

변수가 선언된 블록에서만 유효한 변수

04

함수가 시작 시 생성, Stack에 생성

05

함수가 종료되면 변수도 소멸

06

초기화 전 쓰레기 값을 가지고 있음

07

매개변수도 지역변수

지역변수

1 개요

```
int main()
{
    int a,b;
    char c;

}
```

```
void sub()
{
    int a;

}
```

지역변수

2 유효범위

1 변수가 선언된 블록에서만 유효

a 3

b

c

```
int main()
{
    int a=3,b;
    char c;

}
```

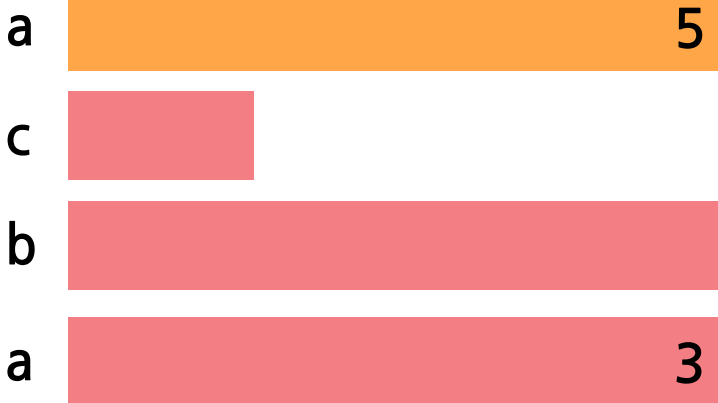
```
void sub()
{
    printf("%d", a);

}
```

지역변수

2 유효범위

2 함수 시작 시 생성되고 종료 시 소멸



```
int main()
{
    int a=3,b;
    char c;

    sub();

}
```

```
void sub()
{
    int a=5;
    printf("%d", a);

}
```

지역변수

2 유효범위

3 생성 시 쓰레기 값을 가짐

4 매개변수도 지역변수임

a 5

b 2

c

b 2

a 3

```
int main()
{
    int a=3,b;
    char c;
    printf("%d", a);
    printf("%d", b);
    b=2;
    sub(a);
    printf("%d", a);
    printf("%d", b);

    return 0;
}
```

```
void sub(int b)
{
    int a=5;
    printf("%d", a);
    b = a+b;
    printf("%d", b);
}
```

지역변수

2 유효범위

C 언어 표준

- 1983년(ANSI : American National Standard Institute)에서 ANSI C라는 표준안을 발표
- 이후 ISO 주도로 표준안을 발표(C99, C11)

5 블록 내에서 변수 선언이 가능(C99)

6 선언된 블록에서 유효

```
for(int i=0;i<5;i++)
{

}
```

```
int main()
{
    int a=3,b=3;
    if (a>0)
    {
        int b=1;
        a +=b;
    }
    printf("%d %d",a, b);
    return 0;
}
```


전역변수

1 개요

01

프로그램 내 전체 함수에서 유효, Global Variable

02

프로그램 시작 시 생성, 데이터 영역에 생성

03

프로그램 종료 시 소멸

04

함수 밖에서 선언

05

자동으로 0으로 초기화

06

전역변수는 프로그램 전체에서 참조하므로 복잡성을 증대

전역변수

1 개요

07

모듈화의 독립성 확보가 어려움

08

메모리 공간을 점유

```
#include <stdio.h>
int sub(int);
int main()
{
    int a
    printf("%d", a);    a=3;
    printf("%d", a);
    a = sub(a);
    printf("%d", a);
    return 0;
}
```

```
int sub( int a)
{
    printf("%d", a);
    a++;
    return a;
}
```

전역변수

1 개요

```
#include <stdio.h>
int a;
int main()
{
    printf("%d", a);    a=3;
    printf("%d", a);
    sub();
    printf("%d", a);
    return 0;
}
```

```
void sub( )
{
    printf("%d", a);
    a++;
}
```

```
#include <stdio.h>
int a;
int main()
{
    int a= 7;
    printf("%d", a);    a=3;
    printf("%d", a);
    sub();
}
```

```
printf("%d", a);
return 0;
}

void sub( )
{
    int a = 5;
    printf("%d", a);
    a++;
}
```

동일한 이름의 전역변수와 지역변수 사용 시 지역변수가 우선함

전역변수

2 다중 파일



extern을 선언하여 외부에서도 참조 가능

extern int 변수;

외부에 선언된 변수를 참조한다는 선언문

main.c

```
#include <stdio.h>
int a;
int main()
{
    int a= 7;
    printf("%d", a); a=3;
    printf("%d", a);
    sub();
    printf("%d", a);
    return 0;
}
```

sub.c

```
extern int a;
void sub( )
{
    printf("%d", a);
    a++;
}
```

학습정리

1. 지역변수



- 변수 스코프(Scope) : 변수 참조가 가능한 유효범위
- 함수 내에 선언된 변수, Local Variable
- 변수가 선언된 블록에서만 유효한 변수
- 함수가 시작 시 생성, Stack에 생성
- 함수가 종료되면 변수도 소멸
- 초기화 전 쓰레기 값을 가지고 있음
- 매개변수도 지역변수
- 유효범위
 - 변수가 선언된 블록에서만 유효
 - 함수 시작 시 생성되고 종료 시 소멸
 - 생성 시 쓰레기 값을 가짐
 - 매개변수도 지역변수임
 - 블록 내에서 변수 선언이 가능(C99)
 - 선언된 블록에서 유효

학습정리

2. 전역변수



- 프로그램 내 전체 함수에서 유효, Global Variable
- 프로그램 시작 시 생성, 데이터 영역에 생성
- 프로그램 종료 시 소멸
- 함수 밖에서 선언
- 자동으로 0으로 초기화
- 전역변수는 프로그램 전체에서 참조하므로 복잡성을 증대
- 모듈화의 독립성 확보가 어려움
- 메모리 공간의 점유
- 다중 파일
 - extern을 선언하여 외부에서도 참조 가능