

C언어 기초 part. 1

Week 1 – 오리엔테이션, C언어의 기본

QnA 메일 : edujongkook@gmail.com

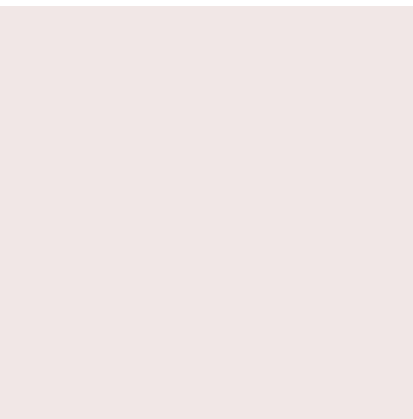
Pdf 파일 : [github.com/edujongkook
/pdf_sbs_c_weekend](https://github.com/edujongkook/pdf_sbs_c_weekend)

목차

A table of contents

- 1 강의 일정, 목표
- 2 Visual studio 개발 환경 설치
- 3 C언어 소개
- 4 변수와 자료형





1.

강의 일정 / 강의 목표



주	교재범위	내용
1-2 주	Chapter 1	c언어의 기본요소 / 데이터 형, 연산자, 수식
3-4 주	Chapter 2	조건문 / 반복문 / 함수 프로그래밍
5-6 주	Chapter 3, 4	포인터와 배열 / 문자열
7-8 주	Chapter 5	포인터 심화, 메모리 동적할당, 구조체

학생부 프로그램 개발

```
C:\Users\origener\Desktop\c_주말\연습_프로젝트\학생부_예제\Debug\학생부_예제.exe
메뉴를 선택하세요
(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)
1
1번째 학생데이터를 추가합니다.
이름 : 김학생
점수 : 80
메뉴를 선택하세요
(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)
3
학생데이터를 출력합니다.
-----
1번째 학생
이름 : 김학생, 점수 : 80
-----
메뉴를 선택하세요
(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)
```

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

typedef struct {
    char* name;
    int score;
}Student;

typedef enum {
    quit,
    add,
    find,
    print,
}Command;

#define STEP_SIZE 3
#define BUFFER_SIZE 100
int max_size = STEP_SIZE;
```

배울게될 개념

구조체

함수

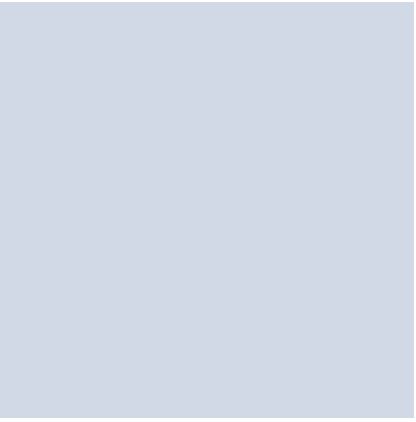
포인터

메모리관리

```
void st_print(Student* st_p, int count);  
void st_find(Student* st_p, int count);
```

```
int main(void) {  
    int count = 0;  
    Student* st_p;  
    st_p = (Student*)malloc(sizeof(Student) * max_size);  
    Command c;  
    while (1) {  
        puts("메뉴를 선택하세요 Wn(0:종료, 1:입력, 2:이름검색, 3:학생부 출력)");  
        scanf_s("%d", &c);  
        switch (c) {  
            case quit:  
                printf("프로그램을 종료합니다.Wn");  
                for (int i = 0; i < count; i++) {  
                    free(st_p[i].name);  
                }  
                free(st_p);  
                return 0;  
            case add:  
                st_add(&st_p, &count);  
                break;  
            case find:
```

변수
연산자
조건문
반복문
배열



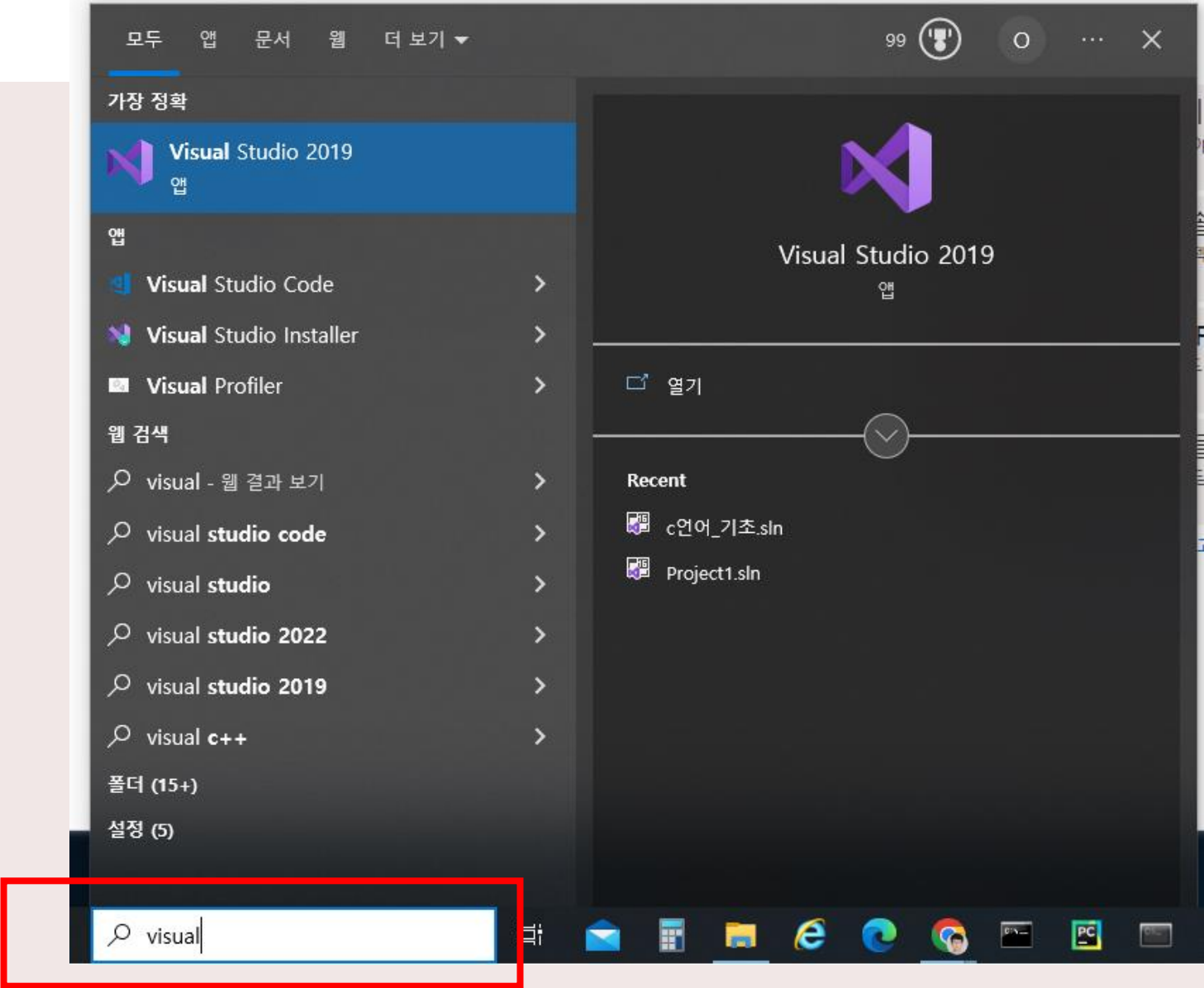
2.

Visual studio 개발환경



Visual studio 실행방법

윈도우 하단 찾기
-> Visual studio



Visual Studio 2019

최근 파일 열기(R)

최근 항목 검색(Alt+S)(S)



▲ 어제



c언어_기초.sln

2023-07-13(목) 오전 11:35

C:\Users\Worigener\Desktop\sbs_c#c언어_기초

시작



리포지토리 복제(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는 .sln 파일 열기



로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집



새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

[코드를 사용하지 않고 계속\(W\) →](#)

새 프로젝트 만들기

최근 프로젝트 템플릿(R)

빈 프로젝트

C++

템플릿 검색(Alt+S)(S)



모든 언어(L)

모든 플랫폼(P)

모든 프로젝트 형식(T)



메이크파일 프로젝트

사용자 빌드 시스템을 가져와 C++를 컴파일하세요.

C++

Windows

데스크톱

콘솔

라이브러리



기본 단위 테스트 프로젝트

기본 Microsoft CppUnitTest 프레임워크를 사용하여 C++ 단위 테스트를 작성합니다.

C++

Windows

테스트



Google Test

Google Test를 사용하여 C++ 단위 테스트를 작성하세요. 사용할 Google Test 라이브러리의 복사본을 포함합니다.

C++

Windows

테스트



내보내기가 있는 동적 연결 라이브러리(DLL)

실행 중인 여러 Windows 앱 간에 공유할 수 있는 .dll을 빌드하세요.

원하는 항목을 찾지 못하셨나요?

[추가 도구 및 기능 설치](#)

뒤로(B)

다음(N)



새 프로젝트 만들기

최근 프로젝트 템플릿(R)

빈 프로젝트

C++

템플릿 검색(Alt+S)(S)



모든 언어(L)

모든 플랫폼(P)

모든 프로젝트 형식(T)



빈 프로젝트

Windows용 C++를 사용하여 처음부터 시작합니다. 시작 파일을 제공하지 않습니다.

C++

Windows

콘솔



콘솔 앱

Windows 터미널에서 코드를 실행합니다. 기본적으로 "Hello World"를 출력합니다.

C++

Windows

콘솔



CMake 프로젝트

.sln 또는 .vcxproj 파일에 종속되지 않은 최신 플랫폼 간 C++ 앱을 빌드하세요.

C++

Windows

Linux

콘솔



Windows 데스크톱 마법사

마법사를 사용하여 고유한 Windows 앱을 만드세요.

C++

Windows

데스크톱

콘솔

라이브러리



Windows 데스크톱 애플리케이션

Windows에서 실행되는 그래픽 사용자 인터페이스를 사용하는 애플리케이션용 프로젝트입니다.

C++

Windows

데스크톱



빈 솔루션

뒤로(B)

다음(N)



3.

C언어 소개



C 언어는 작고 빠르다.

대부분의 환경에서 잘 동작한다.

배우기는 어렵지만 배울 문법이 많이 없다.

사용자 보다는 하드웨어와 가깝다.

개발에 도움을 주는 보조기능들이 별로 없다.





TOUCH SENSITIVE



NETWORK STORAGE



SMARTPHONE OS



FRAMEWORK



CODE TEMPLATE



DOWNLOAD SPEED



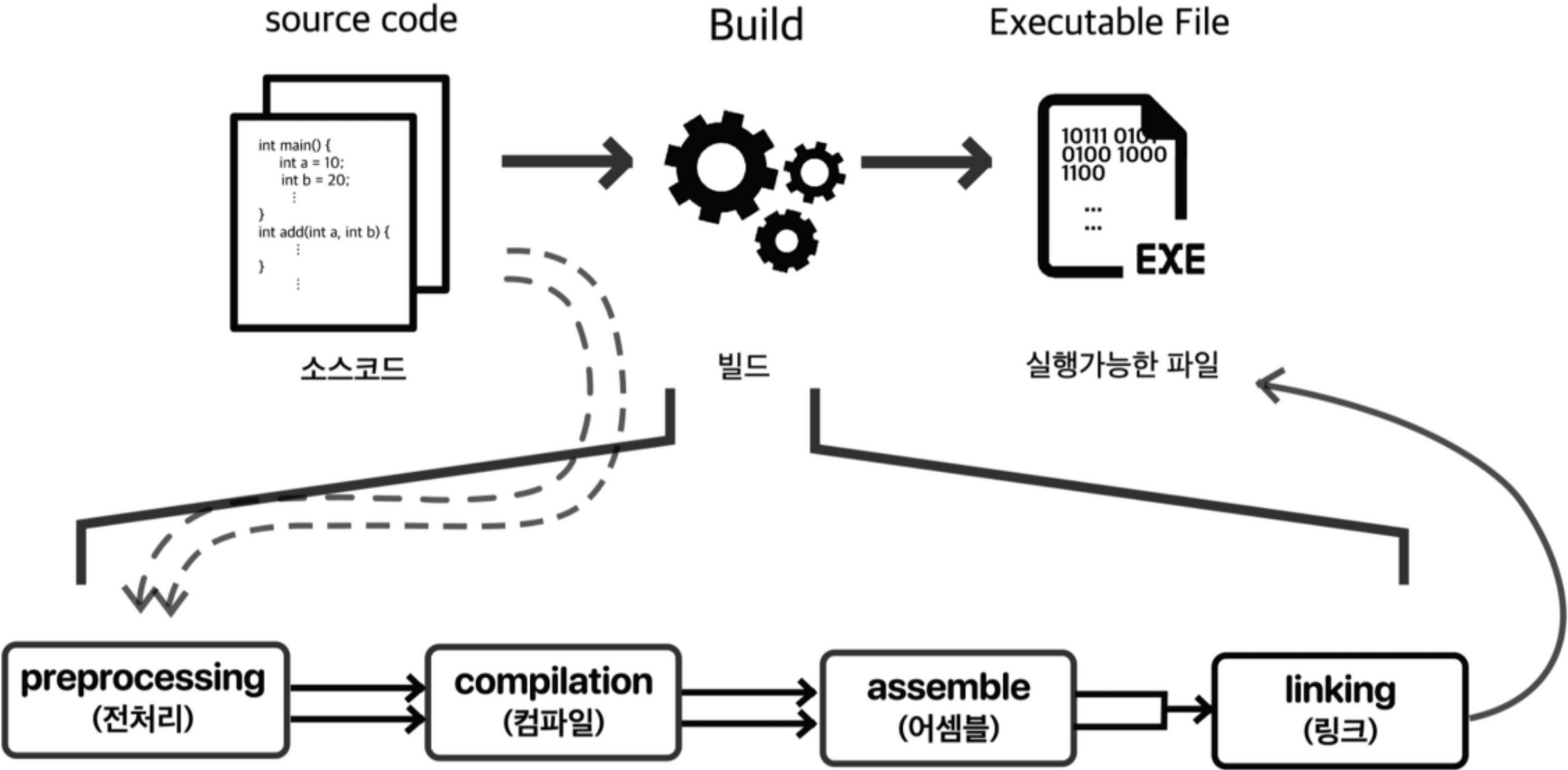
DATA TRANSFER

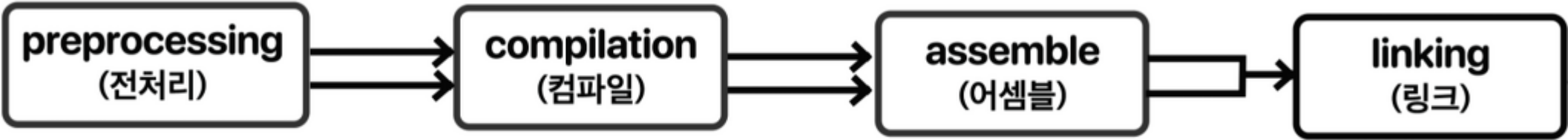
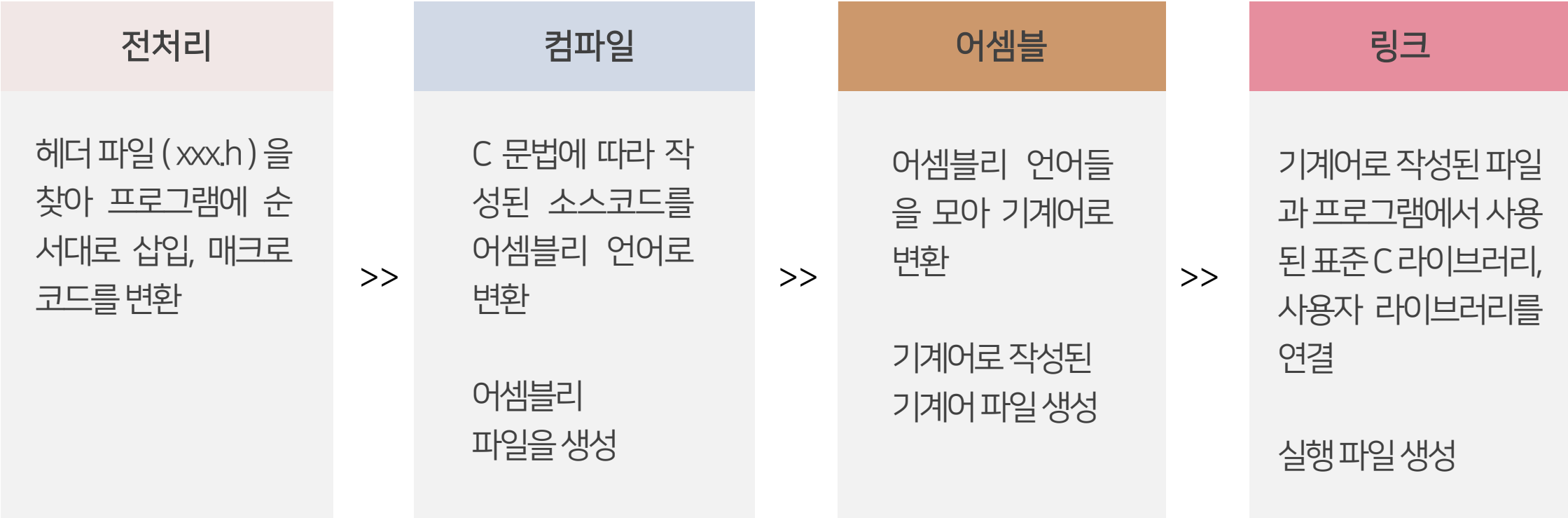


SIGNAL



VIDEO EDITING

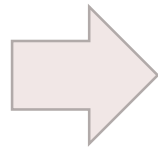




```
#include <stdio.h>

int main(void) {
    printf("hello!");
    return 0;
}
```

c언어



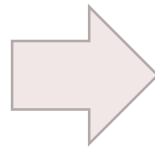
컴파일 과정

```
mov eax, 1
push eax
push DWORD "hello!"
call printf
```

어셈블리 언어

```
mov eax, 1  
push eax  
push DWORD "hello!"  
call printf
```

어셈블리 언어



어셈블 과정

```
110100100100000010000001111  
110000000000011110111100000  
000111110110011111110000000
```

기계어

```
#include <stdio.h>

int main(void) {
    printf("hello!");
    return 0;
}
```

#include 는 외부의 h 파일을 소스코드에 포함시키는 특수명령입니다.

h 파일은 header(헤더) 파일로 외부의 c언어 코드에 대한 정보가 담겨있는 파일입니다.

stdio.h 는 standard input output 의 약자로 C언어 표준 입력출력과 관련된 기능을 제공합니다.

```
#include <stdio.h>

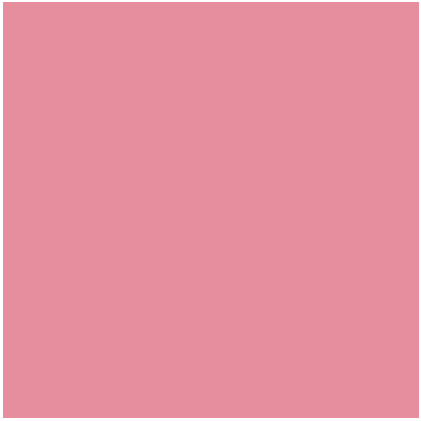
int main(void) {
    printf("hello!");
    return 0;
}
```

int main(void) 는 함수라는 형태로 추후 다시 배우게됩니다.

main() 은 특수한 함수로 프로그램이 실행될 때 가장먼저 시작되는 함수 입니다.

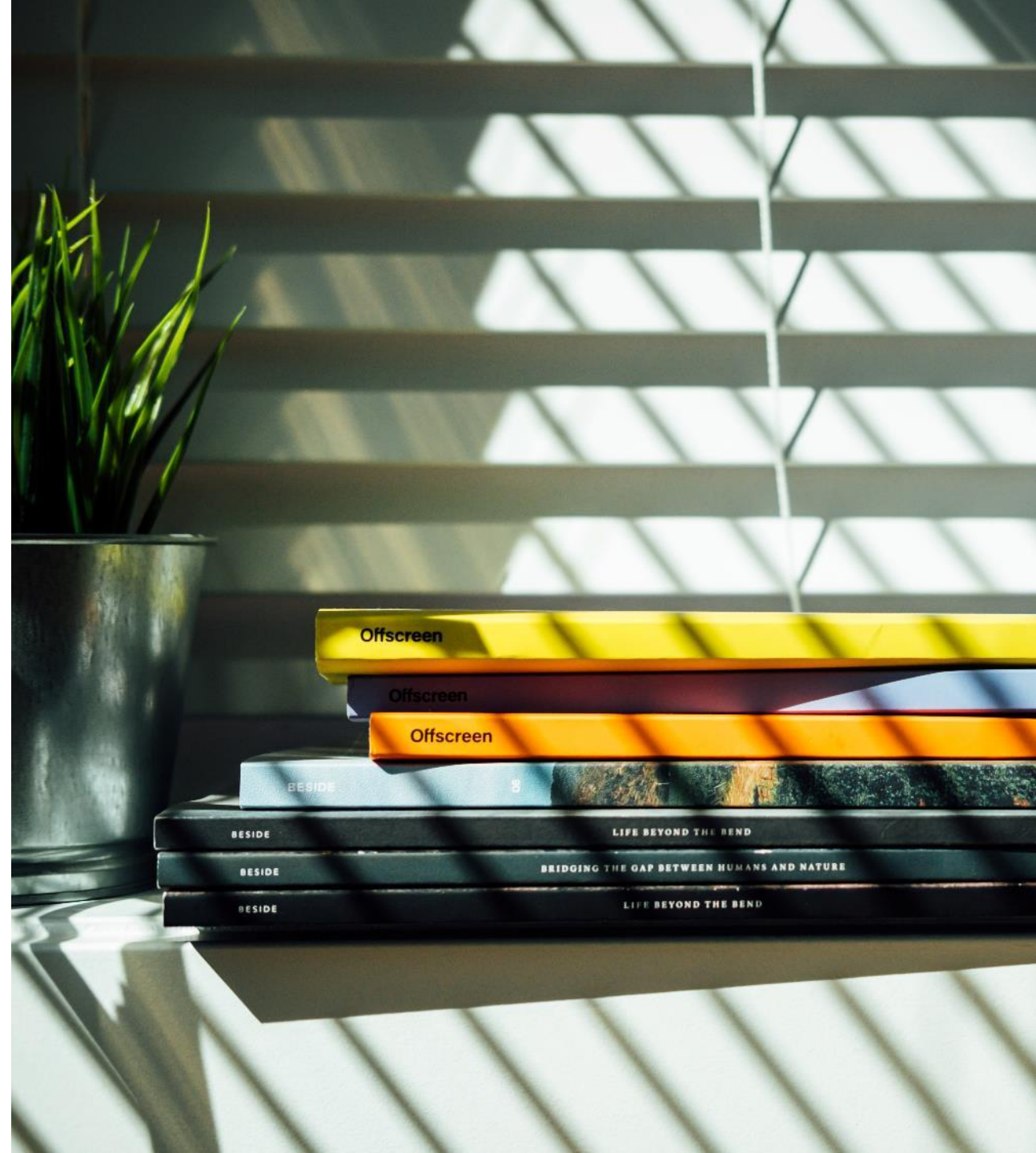
```
/*
파일명 : 연습.c
작성자 : 김종국
수정 날짜 : 2024-01-20
기능 : printf 함수를 통한 문자열 출력
*/
#include <stdio.h>

int main(void) {
    printf("hello!"); // hello 를 출력합니다.
    return 0;
}
```



4.

변수와 자료형





CPU
(Central Processing Unit)

1101001001000000100000001111
1100000000000111101111000000
0001111101100111111110000000



RAM
(Random Access Memory)

```
int main(void) {  
    int number;  
    return 0;  
}
```

변수의 선언

```
int number;
```

자료형 변수이름

int 는 정수(integer)형 자료형을 의미합니다.

```
int main(void) {  
    int number;  
    number = 10;  
    return 0;  
}
```

변수의 초기화

```
number = 10;
```

변수이름 = 값

변수 number에 값 10을 대입한다고 표현합니다.

```
int main(void) {  
    int number = 10;  
    return 0;  
}
```

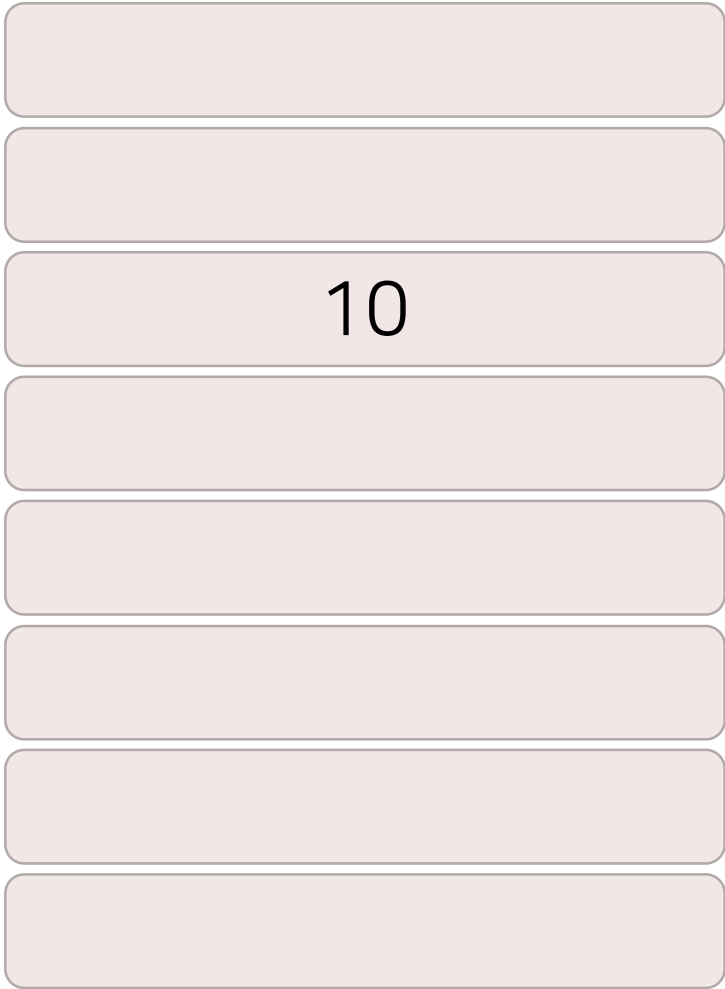
선언과 동시에 초기화

```
int main(void) {  
    int number = 10;  
    number = 50;  
    return 0;  
}
```

변수에 새로운 값 대입



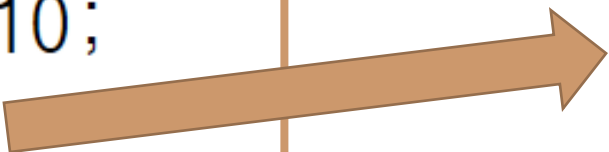
```
int main(void) {  
    int number = 10;  
    return 0;  
}
```



메모리



```
int main(void) {  
    int number = 10;  
    number = 50;  
    return 0;  
}
```



메모리

기존 변수에 새로운 값을 넣으면 기존의 값은 사라지고 변수의 위치는 변하지 않은 상태에서 값만 새롭게 바뀝니다.



변수를 사용하는 이유

변수를 사용하는 이유는 값을 메모리에 저장하기 위해서 입니다. 코딩을 할 때 값을 저장하지 못한다면 매번 값을 개발자가 기억하거나 별도로 기록해야 하므로 코딩을 하기 어려워 집니다.



메모리

```
int main(void) {  
    int number1, number2;  
    number1 = 1;  
    number2 = 2;  
    int number3 = 3, number4 = 4;  
  
    return 0;  
}
```

동시에 여러변수를 선언,
초기화 할 수 있지만
권장되지는 않습니다.
가독성이 떨어지기 때문입니다.


```
int main(void) {  
    /* 사용 가능한 변수명 */  
    int num = 3;  
    int number5 = 5;  
    int num_ber = 10;  
    int Number = 15;  
  
    /* 사용 불가능 (컴파일 에러) */  
    int hello boy = 3; // 공백 포함  
    int hi#$ = 5; // 특수 기호  
    int 8number = 8; // 숫자로 시작  
    int short = 3; // 특수기능 키워드  
}
```

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

```
int main(void) {
```

```
    int number1 = 3;
```

```
    printf("%d\n", number1);
```

```
    int number2;    // 먼저 선언만 하고
```

```
    number2 = 5;    // 나중에 값을 할당
```

```
    printf("%d\n", number2);
```

```
    return 0;
```

```
}
```

%d는 decimal (십진수)
의약자입니다.

\n은 출력할 때 엔터를
친 것처럼 칸을
새로 만듭니다.

자료형	형태	크기	값의 범위
char	정수(문자)	1 바이트	-128 ~ 127
short	정수	2 바이트	-32,768 ~ 32,767
int		4 바이트	-2,147,483,648 ~ 2,147,483,647
long		4 바이트	-2,147,483,648 ~ 2,147,483,647
long long		8 바이트	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807
float	실수	4 바이트	1.175494e-38 ~ 3.402823e+38
double		8 바이트	2.225074e-308 ~ 1.797693e+308
long double		8 바이트 이상	2.225074e-308 ~ 1.797693e+308

문자	알파벳 문자 및 기호: 'A', '+', '&' ...	char
정수	양의 정수, 0, 음의 정수: 10, 100, -50 ...	short int long long long
실수	소수점 이하 값을 가진 숫자: 3.14, 0.01, -5.5 ...	float double long double

```
#include <stdio.h>

int main(void) {
    int number = 10;
    float pi = 3.14;
    printf("number : %d\n", number);
    printf("pi : %f\n", pi);
    return 0;
}
```

실수형태의 자료형 float을
출력 할때는 %f 를 사용합니다.

```
#include <stdio.h>

int main(void) {
    char ch = 'A';
    char str[] = "abc";
    printf("ch : %c\n", ch);
    printf("str : %s\n", str);
}
```

%c 는 character(문자) 의 약자입니다.

%s 는 string(문자열) 의 약자입니다.

C에서 문자열은 char 자료형 변수를 배열
또는 포인터를 사용하여 표현합니다.

str[] 은 배열입니다. (추후 다시 다룹니다)

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

```
int main(void) {  
    char ch = 'A';  
    int num = 10;
```

```
    printf("size of ch : %d\n", sizeof(ch));  
    printf("size of num : %d\n", sizeof(num));
```

```
}
```

자료형	크기
char	1 바이트
short	2 바이트
int	4 바이트
long	4 바이트
long long	8 바이트
float	4 바이트
double	8 바이트
long double	8 바이트 이상

```

int main(void) {
    char ch1 = 66, ch2 = 'B';
    short sh1 = 67;
    int in1 = 68;

    printf("%c\n", ch1);
    printf("%c\n", ch2);
    printf("%c\n", sh1);
    printf("%c\n", in1);
    return 0;
}

```

065	A	089	Y	113	q
066	B	090	Z	114	r
067	C	091	[115	s
068	D	092	\	116	t
069	E	093]	117	u
070	F	094	^	118	v
071	G	095	_	119	w
072	H	096	`	120	x
073	I	097	a	121	y
074	J	098	b	122	z
075	K	099	c	123	{
076	L	100	d	124	
077	M	101	e	125	}
078	N	102	f	126	~
079	O	103	g		