

WONDER



놀랄 만큼 재미있는 C언어 기초

조성호 지음

INFINITYBOOKS
인피니티북스

C프로그램 구성요소

학습목차

- ▶ 3.1 C프로그램 구조
- ▶ 3.2 함수와 라이브러리
- ▶ 3.3 변수의 선언과 활용

3장은 C프로그램의 기본 구조를 설명하고
각 구성요소의 특징과 주의 사항을 볼 거야
함수 및 라이브러리, 변수에 대해 알아볼 수 있어



즉, 중요한 장이라는 거지!



3장을 허술하게
여기저기 마세요 꼬옥!



저도 강조!



드디어 뭔가
나오나보다

까꿍^^



1 C코드 구조

- 이메일에도 구조와 절차가 있듯이 C코드에도 지켜야 할 구조와 절차가 있음

이메일 주소만 가지고선
누가 보냈는지 알 수 없죠.
학번과 이름이 필요해요.

교수님의 담당 과목이
여러 개라면 쿵~
집어 [C언어 과제]라고
쓰면 좋아요 ^^

본문은 맞춤법을 정확히
쓰고 띄어쓰기를 하여
읽기 좋게 만들자!

첨부파일은 있는 경우가
있고 없는 경우도 있다.

받는사람	prof@infinity.ac.kr
보내는사람	kyw1456@naver.com
제목	[C언어 과제] 284154 김영웅 과제제출
본문	<p>학번 284154, 김영웅입니다. 좋은 수업해주셔서 항상 감사드립니다. 이번 금요일 제출마감인 C언어 과제를 제출합니다. return.c와 return.exe를 zip으로 압축하여 보냅니다.</p>
첨부파일	C_284154_김영웅.zip

도대체누가 보낸거야?
성의 없는 이메일에는
성의 없는 과제점수가
그 보답이다!

받는사람	prof@infinity.ac.kr
보내는사람	rty1234@naver.com
제목	과제
본문	<p>C언어과제제출</p>
첨부파일	C.zip

- main() 함수는 이메일의 "본문"에 해당 - C코드에는 main() 함수가 꼭 있어야 함
- C코드에서 작업의 시작과 끝을 알리는 것은 중괄호 {}
- printf("Hello World")는 Hello World 글자를 화면에 출력하는 함수
- printf() 함수를 사용하기 위해서는 #include <stdio.h> 필요
- return 0은 정상적으로 종료한다는 의미

```
1  #include<stdio.h>
2
3  int  main()
4  {
5      printf("Hello World");
6      return 0;
7  }
```

라이브러리

main() 함수

작업 시작, 끝

종료, 0 반환

- C코드를 만들 때 사용되는 단어를 키워드(keyword)라 부름
- 키워드 keyword 란 C언어 문법에서 중요(key)하게 사용되는 단어(word)

표 3-1 C코드 키워드

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

2 C코드 기본 규칙

- 일반적으로 코드의 맨 끝에는 세미콜론(;)을 붙임
- 샵(#)으로 시작하는 코드에는 세미콜론(;)을 붙이지 않음
- C코드를 작성할 때, 띄어쓰기와 들여쓰기를 꼭 해야 함

	띄어쓰기 올바른 예	띄어쓰기 오류
01	<code>int main()</code>	<code>intmain()</code>
02	<code>return 0;</code>	<code>return0;</code>

	띄어쓰기 예	띄어쓰기 다른 예
01	<code>#include<stdio.h></code>	<code>#include < stdio.h ></code>
02		
03	<code>int main()</code>	<code>int main () {</code>
04	<code>{</code>	<code>printf ("Hello World") ;</code>
05	<code>printf("Hello World");</code>	<code>return 0 ;</code>
06	<code>return 0;</code>	<code>}</code>
07	<code>}</code>	

- 들여쓰기를 꼭 해야 함

	들여쓰기 함	들여쓰기 안 함
01	<code>#include <stdio.h></code>	<code>#include <stdio.h></code>
02		
03	<code>int main()</code>	<code>int main()</code>
04	<code>{</code>	<code>{</code>
05	<code> printf("Hello World");</code>	<code>printf("Hello World");</code>
06	<code> return 0;</code>	<code>return 0;</code>
07	<code>}</code>	<code>}</code>

3 주석이란?

- 주석은 일종의 메모

```
01  /* 전산과 284154 김영웅
02     C프로그래밍 과제물 */
03
04  /*****
05     2021. 6. 7.
06     처음으로 만든 과제물
07     *****/
08  #include <stdio.h>
09
10  int main()
11  {
12      printf("Hello World");           // 화면에 Hello World 출력하는 함수
13      return 0;                       // 어머니(운영체제)에게로 돌아간다.
14  }
```


1 함수란?

- 모듈이란 일정한 기능을 수행하도록 조립된 부품 덩어리
- 함수(function)란 정해진 절차에 따라 데이터를 처리하는 모듈



2 라이브러리와 헤더파일

- C언어에는 마음대로 사용할 수 있는 수천개의 함수가 있음
- 미리 만들어 놓은 함수를 라이브러리 혹은 라이브러리 함수라 부름
- 관련 있는 함수들을 모아 놓은 파일을 헤더파일(header file), 확장자는 .h

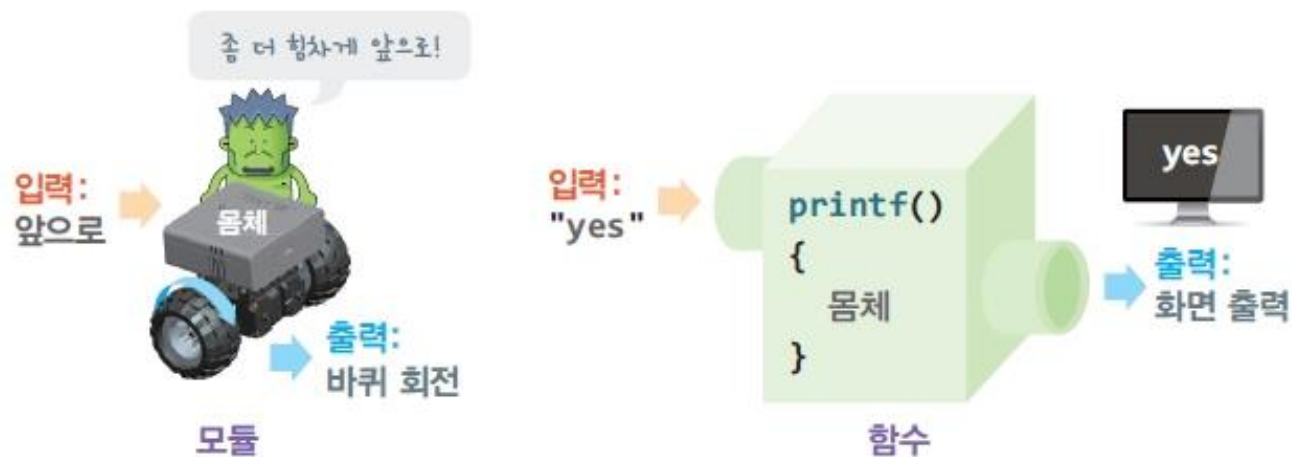


- C코드에서 가장 많이 사용하는 헤더파일은 `stdio.h`
- 수학 관련 라이브러리 함수들이 모여 있는 헤더파일의 이름은 `math.h`



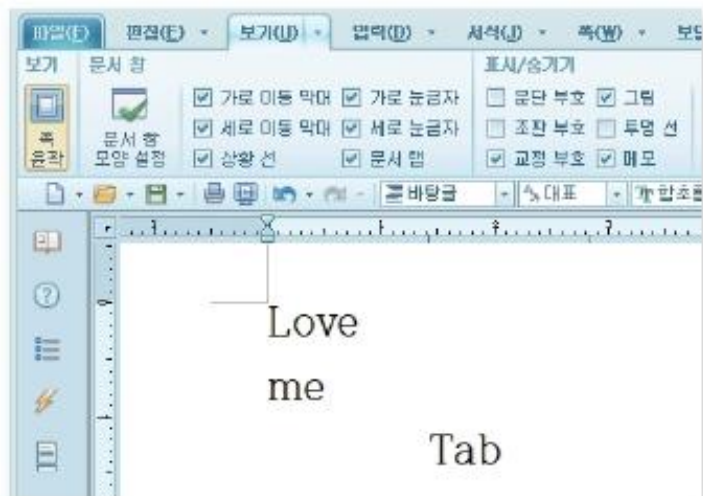
3 함수 구조

- 모듈은 입력, 출력, 몸체로 구성
- 함수이름(입력) { 몸체 }



4 Printf() 함수와 제어문자

- 제어문자(control character)란 화면에는 출력되지 않지만, 화면의 상황을 제어하는 문자



- 줄바꿈 제어문자는 \n

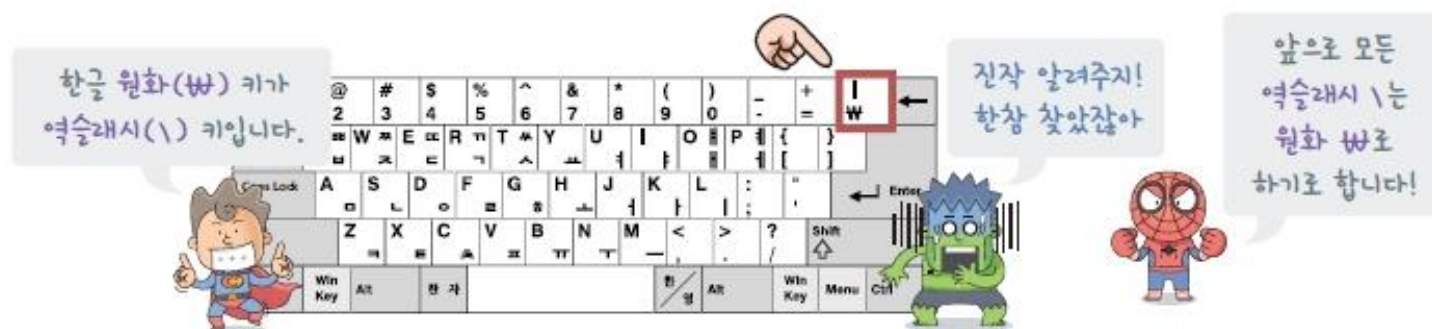
```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("Hello World\n");           // 줄 바꿈 제어문자 \n
06      printf("I am a C programmer");
07
08      return 0;
09  }
```

실행 화면

```
Hello World
I am a C programmer
```

```
05  printf("Hello World\nI am a C programmer");
```

- 키보드에서 역슬러시 위치



1 변수란?

- C코드에서 데이터를 담는 그릇이 변수(variable)

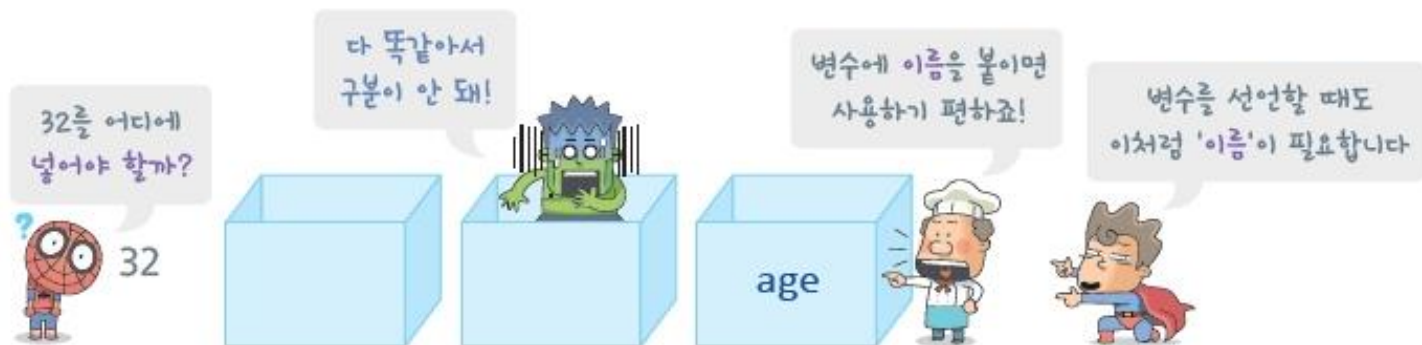


> 변수와 3가지 자료형

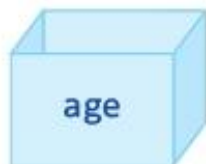


2 변수 만들기

> 변수 선언

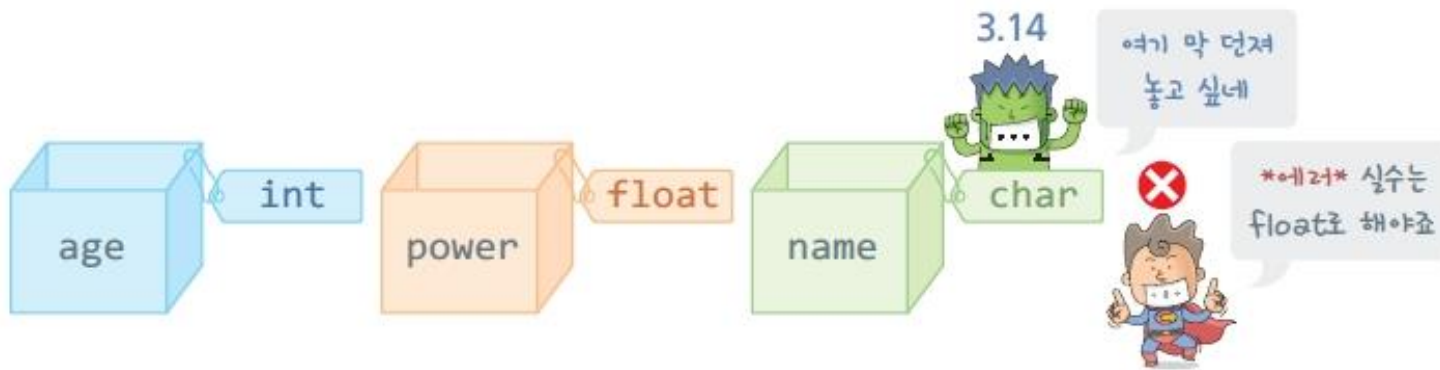


> 변수 선언 방법



```
01 int age;
02 float power;
03 char name;
```

정수형 변수 age
실수형 변수 power
문자형 변수 name



> 변수 이름 규칙

표 3-2 변수이름 규칙

규칙	사용 가능	사용 불가능	사용 불가능 이유
영문자, 숫자, 밑줄 가능	<code>int age_p1;</code>	<code>int age p1;</code>	공백 대신 밑줄 사용
숫자부터 시작하면 안 됨	<code>int num3;</code>	<code>int 3num;</code>	숫자를 뒤에 쓸 것
키워드 사용 불가	<code>int int_sum;</code>	<code>int int;</code>	키워드를 포함한 단어 가능
밑줄부터 시작 가능	<code>int _num3;</code>	<code>int num!;</code>	특수문자 사용 불가능

3

대입 연산자

- 오른쪽 값을 왼쪽에 대입

```
int age;  
age = 24;
```

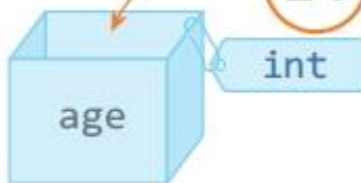
정수형 변수 age 선언
변수 age에 24 대입

대입 연산자 =은 오른쪽
값을 왼쪽에 넣는 거예요



age = 24 ;

대입 연산자



24

int

age

age와 24가
같다는 거잖아!
쉽네



노노!
'같다'는
==임요!



> 다양한 대입 연산자 사용 방법

```

01  int age1;
02  int age2;
03
04  age1 = 24;
05  age2 = 32;
06
07  age1 = age2;

```

```

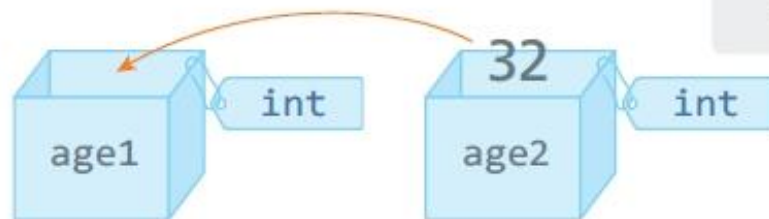
int age1, age2;

age1 = 24;
age2 = 32;

age1 = age2;

```

age1 = age2;
대입 연산자



대입 연산자 =은
오른쪽 변수 age2 값 32를
왼쪽에 넣는 거예요

정수 초기화 예

```
int age1 = 24, age2 = 32;
```

실수 초기화 예

```
float power1 = 24.0, power2 = 32.0;
```

4 변수 값 출력하기

> 서식문자

```
01  #include <stdio.h>
02
03  int main()
04  {
05      printf("내 나이 = %d 친구 나이 = %d", 29, 31);    // %d는 나열 연산자
06
07      return 0;
08  }
```

실행 화면

내 나이 = 29 친구 나이 = 31

첫 번째 값 정수 29는
이 자리에 출력되는 거예요

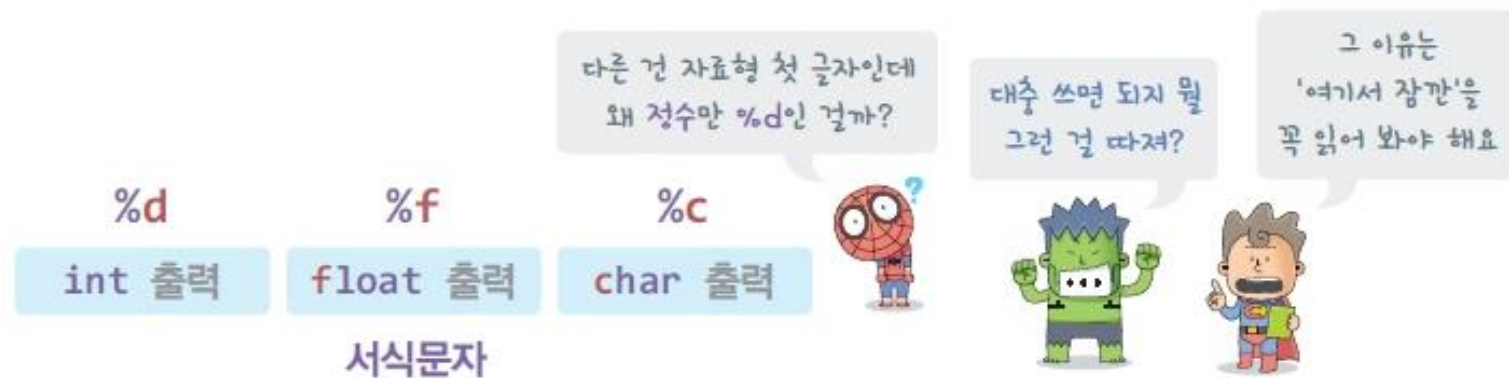


두 번째 값 정수 31은
이 자리에 출력되죠



printf(" 내나이 = %d 친구나이 = %d " , 29, 31);

> 서식문자 종류



> 서식문자 예러

01	<code>printf("내 나이 = %d");</code>	%d에 대응하는 데이터 없음
02	<code>printf("내 나이 = %d", 29, 31);</code>	31에 대응하는 %d 없음
03	<code>printf("내 나이 = %d 친구 나이 = %d", 29);</code>	두 번째 %d에 대응하는 데이터 없음

[예제 3-5] 여러 변수 출력 코드

```
01  #include <stdio.h>
02
03  int main()
04  {
05      int num1, num2;                // 같은 종류의 변수는 쉼표로 연결
06
07      num1 = 21;
08      num2 = 17;
09
10      printf("num1 = %d, num2 = %d\n", num1, num2);
11      num1 = num2;                  // num2 값 num1에 대입
12      printf("num1 = %d, num2 = %d\n", num1, num2);
13
14      return 0;
15  }
```

실행 화면

num1 = 21, num2 = 17

num1 = 17, num2 = 17

5 대입 연산자 활용

[예제 3-6] 대입 연산자 활용 코드

```
01  #include <stdio.h>
02
03  int main()
04  {
05      int num1 = 21, num2 = 17, num3 = 0;
06
07      num3 = 9 + 5;
08      printf("num3 = %d\n", num3);
09      num3 = num2 + 5;          // num2 값에 5 더하여 num3에 대입
10      printf("num3 = %d, num2 = %d\n", num3, num2);
11      num3 = num1 + num2;      // num1 값에 num2 값 더하여 num3에 대입
12      printf("num3 = %d\n", num3);
13
14      return 0;
15  }
```

실행 화면

```
num3 = 14
num3 = 22, num2 = 17
num3 = 38
```

LAB 1 ▶ 사칙 연산 코드 만들기

```
01  #include <stdio.h>
02
03  int main()
04  {
05      int num1 = 7, num2 = 4;
06      float num3 = 7 / 4;           // 나눗셈 결과는 실수
07
08      printf("num1 = %d, num2 = %d\n", num1, num2);
09      printf("덧셈 = %d\n", num1 + num2);
10      printf("뺄셈 = %d\n", num1 - num2);
11      printf("곱셈 = %d\n", num1 * num2);
12      printf("나눗셈 = %f\n", num3);
13
14      return 0;
15  }
```

실행 화면

```
num1 = 7, num2 = 4
덧셈 = 11
뺄셈 = 3
곱셈 = 28
나눗셈 = 1.000000
```

LAB 2 ▶ 루트 값 구하기

```
01  #include <stdio.h>
02
03
04  int main()
05  {
06      float num1;
07
08      num1 = sqrt(7);
09      printf("루트 %d = %f\n", 7, num1);
10
11      return 0;
12  }
```

실행 화면

루트 7 = 2.645751