

WEEK 02

자료형과 변수

학습목표

- I. C 프로그램의 기본 개념 이해
- II. 자료형을 이해하고, 자료형에 따른 변수 선언과 이용 이해
- III. 상수의 기본 개념 이해

학습목차

- I. 제 1교시 프로그래밍 기초
- II. 제 2교시 자료형과 변수선언
- III. 제 3교시 상수 표현 방법

The background of the slide is a solid blue color with a pattern of overlapping, semi-transparent geometric shapes, primarily triangles and diamonds, in various shades of blue and purple. On the left side, there is a vertical strip of images. The top part shows a hand holding a white bar chart with several bars of increasing height. Below this, there are images of interlocking gears and a close-up of a computer keyboard, all with a blue tint.

I. 프로그래밍 기초

1. 주석과 들여쓰기
2. 키워드와 식별자
3. 기초를 다지기 위한 사례

1. 주석과 들여쓰기

◆ 문장과 주석

❖ 문장

- 프로그램에서 컴퓨터에게 명령을 내리는 최소 단위
 - 세미콜론 ;으로 종료

❖ 주석

- 프로그램 내용에는 전혀 영향을 미치지 않는 설명문
- 주석 /* ... */은 중첩하여 사용 불가능

블록 주석 {

```

/* ← 주석 시작
   솔루션 / 프로젝트 / 소스파일: Ch02 / Prj01 / comments.c
   C 프로그램의 기초를 다지기 위한 주석, 문장, 키워드 등 이해
   V 1.0 2021. 06. 29 강환수 작성
*/ ← 주석 종료
#include <stdio.h>

// 운영체제가 호출하는 함수, 매개변수(없음) ← 한 줄 주석
int main(void)
...
  
```

주석 시작인 /*와 * 사이에 공백이 없어야 하며, 마찬가지로 주석 종료 표시인 */와 * 사이에도 공백이 없어야 한다.

[출처] 강환수 외, Perfect C 3판, 인피니티북스

1. 주석과 들여쓰기

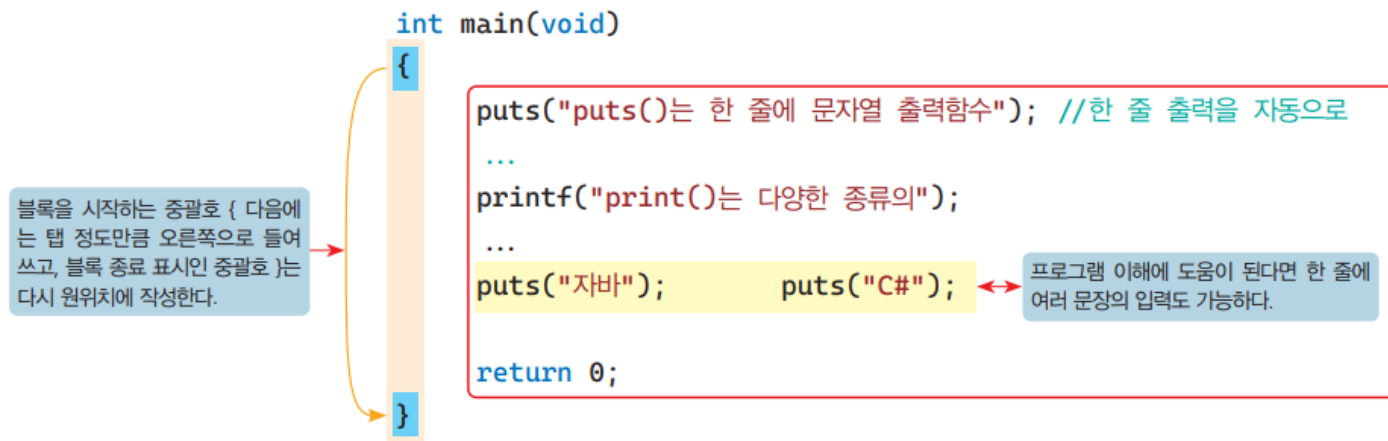
◆ 블록과 들여쓰기

❖ 블록

- 블록(block)은 중괄호로 { ... } 나타내며 여러 문장의 집합을 의미

❖ 들여쓰기

- 블록 내부에서 문장들을 탭(tab) 정도만큼 오른쪽으로 들여 쓰는 소스 작성 방식



1. 주석과 들여쓰기

◆ 소스의 모양

❖ 줄 구분, 들여쓰기

- 들여쓰기 하지 않고 프로그램을 작성해도 오류가 발생하지 않음
- 오른쪽 소스는 상대적으로 프로그램의 이해력을 떨어뜨림
- 왼쪽 소스와 같이 적절한 줄 구분과 빈 줄 삽입, 들여쓰기는 프로그램의 이해력을 돕는데 매우 중요한 요소

들여쓰기가 잘된 소스

```
#include <stdio.h>

int main(void)
{
    puts("C언어");

    return 0;
}
```

들여쓰기가 잘못된 소스

```
#include <stdio.h>
int main(void) { puts("C언어");
return 0;
}
```

↔
오른쪽은 왼쪽소스와
비교하여 상대적으로
이해하기 어려운
소스이다.

2. 키워드와 식별자

◆ 키워드

❖ 키워드, 예약어

- 프로그래밍 언어에서 문법적으로 의미 있는 단어로 사용하기 위해 미리 정의해 놓은 단어
- ANSI에서 지정한 C 언어의 32개의 기본 키워드

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

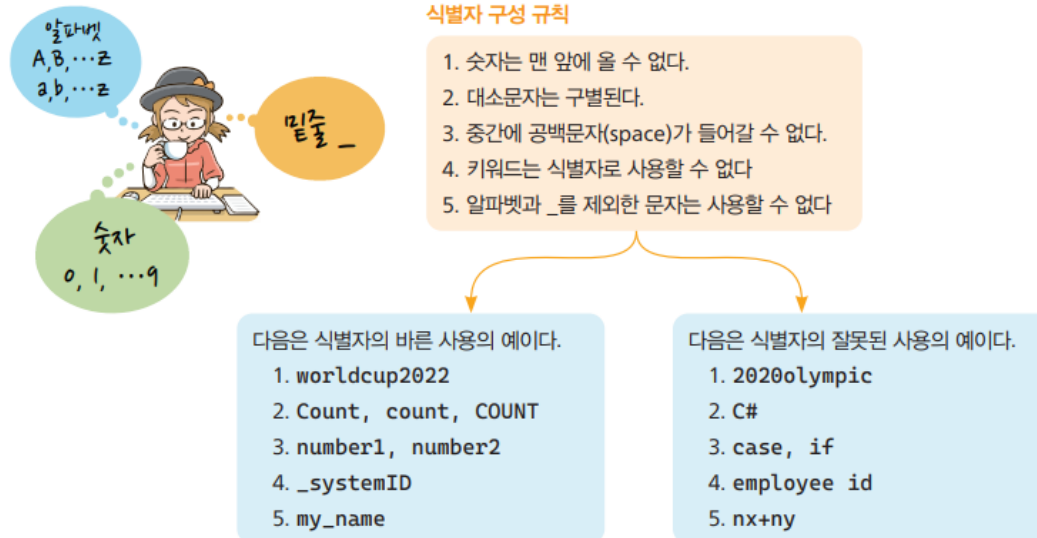
이러한 단어가 C에서 사용되는 기본 키워드로 분법적인 고유한 의미가 있다.

2. 키워드와 식별자

◆ 식별자

❖ 프로그래머가 정의하여 사용하는 단어

- 변수 이름과 함수 이름으로 age, year, puts, main, printf 등
- 영문자(대소문자 알파벳), 숫자(0 ~ 9), 밑줄(_)로 구성
- 첫 문자로 숫자가 나올 수 없음
- 식별자는 서로 구별되어야 하며 키워드는 식별자로 이용 불가능



3. 기초를 다지기 위한 사례

◆ 키워드와 식별자 그리고 주석 등을 이해하기 위한 프로젝트

실습예제 3-1	Prj01	01comments.c	C 언어 기초를 다지기 위한 주석, 문장, 키워드 등 이해	난이도: ★
	<pre>01 /* 02 솔루션 / 프로젝트 / 소스파일: ch03 / Prj01 / 01comments.c 03 C 프로그램의 기초를 다지기 위한 주석, 문장, 키워드 등 이해 04 V 1.0 05 */ 06 #include <stdio.h> 07 08 // 운영체제가 호출하는 함수, void로 매개변수 없음을 표시 09 int main(void) 10 { 11 puts("3장 첫 C 프로그램!\n"); 12 // 13 printf("키워드: int void return 등\n"); 14 printf("식별자: main puts printf 등\n"); 15 // 16 return 0; 17 }</pre> <div data-bbox="1141 596 1556 682"><p>여러 줄에 걸친 블록 주석으로 비주얼 스튜디오에서 주석은 모두 초록색으로 표시</p></div>			
결과	<p>3장 첫 C 프로그램!</p> <p>키워드: int void return 등</p> <p>식별자: main puts printf 등</p>			



1. 프로그래밍 기초

1교시 수업을 마치겠습니다.

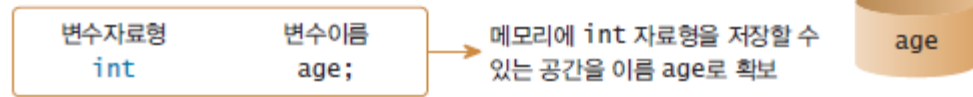
Ⅱ. 자료형과 변수선언

1. 변수선언
2. 변수의 3요소와 이용
3. 기본 자료형

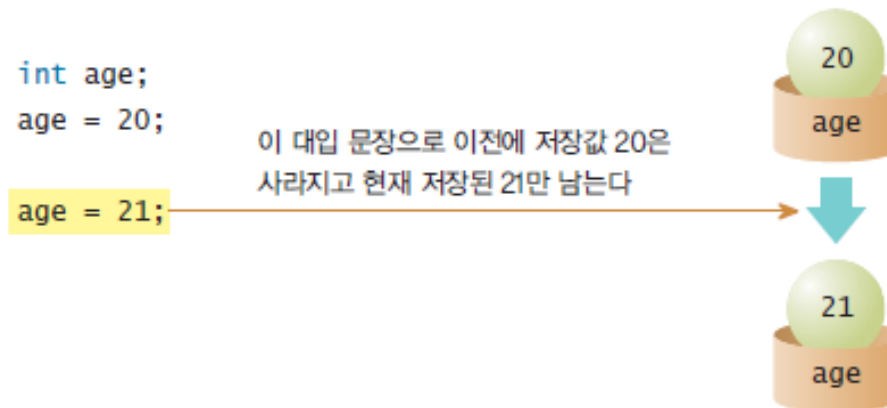
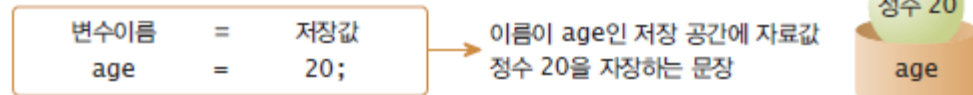
1. 변수선언

- ❖ 변수(variables)는 자료값을 저장하는 공간
- ❖ 변수 선언
 - 변수 자료형과 변수 이름을 나열하여 표시

변수선언문



대입문

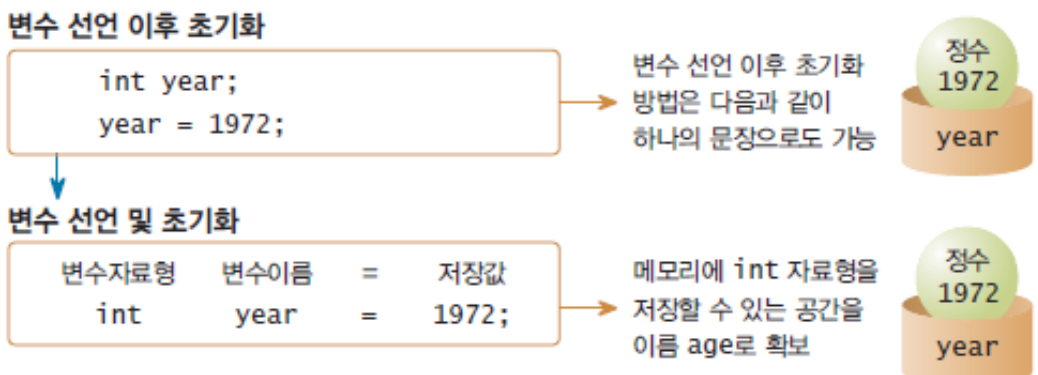


1. 변수선언

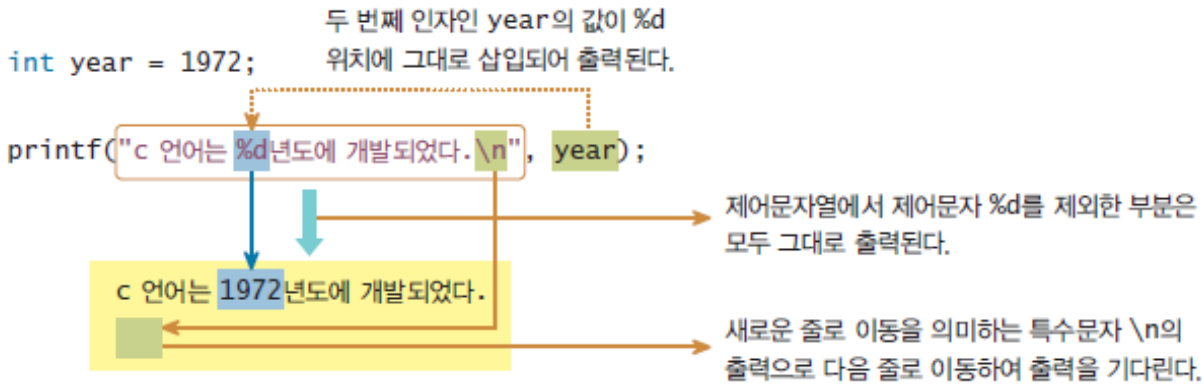
◆ 변수 초기화와 제어 문자

❖ 변수 초기화

- 선언한 이후에는 반드시 값을 저장하도록



❖ 제어문자



2. 변수의 3요소와 이용

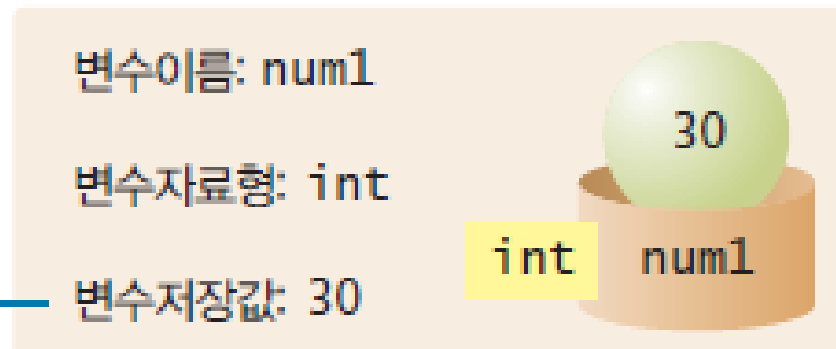
◆ 변수 3요소

- ❖ 변수 이름
- ❖ 변수 자료형
- ❖ 변수 저장값

```
int num1 = 30
```

변수 저장값은
대입 문장에 의해
계속 바뀔 수 있다.

변수 3요소



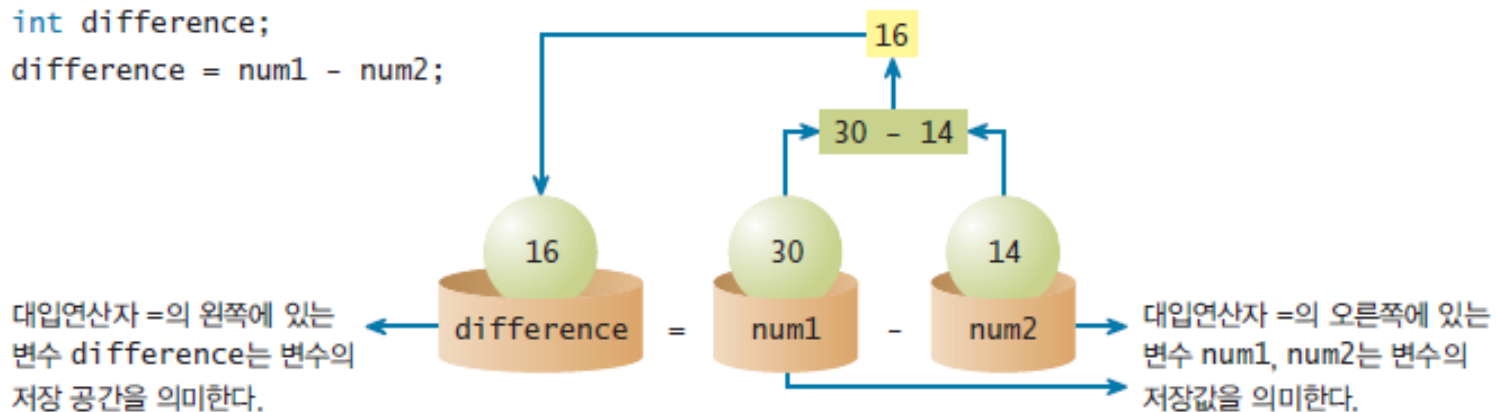
2. 변수의 3요소와 비용

◆ 변수 사용의 의미 2가지

❖ 변수 저장 장소와 값

- 대입연산자 =의 왼쪽에 위치한 변수
 - 저장공간의 사용을 의미
- 대입연산자 =의 오른쪽에 위치한 변수
 - 저장값의 사용을 의미

```
int num1 = 30, num2 = 14;
int difference;
difference = num1 - num2;
```



2. 변수의 3요소와 비용

실습예제 3-3	Prj03	03addsub.c	변수의 l-value와 r-value의 이해와 변수의 값을 더하고 빼기	난이도: ★
<pre> 01 /** 02 * 소스: 03addsub.c 03 * 버전: V 1.0 04 **/ 05 06 #include <stdio.h> 07 08 int main(void) 09 { 10 int data1 = 20, data2 = 13; 11 12 //대입 연산자의 왼쪽과 오른쪽에서의 변수의 의미 해석 13 int diff = data1 - data2; 14 int sum = data1 + data2; 15 16 printf("data1: %d, data2: %d\n", data1, data2); 17 printf("차: %d, 합: %d\n", diff, sum); 18 19 return 0; 20 } </pre>				
결과	<div> <div>data1: 20, data2: 13</div> <div>차: 7, 합: 33</div> </div>			

변수 diff는 l-value로 변수 자체를 의미
하고, data1과 data2는 r-value로 변
수에 저장된 값이 연산에 참여

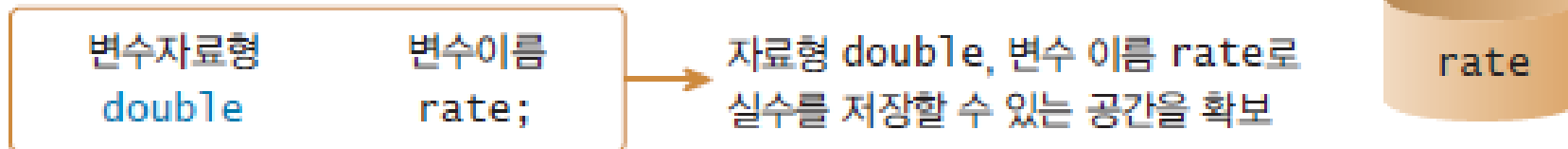
3. 기본 자료형

◆ 자료형 개요

❖ 자료의 종류

- 변수의 저장공간 크기와 저장되는 자료값의 종류가 결정
 - 문자형(character type)
 - 정수형(integer type)
 - 부동소수형(floating point data type)

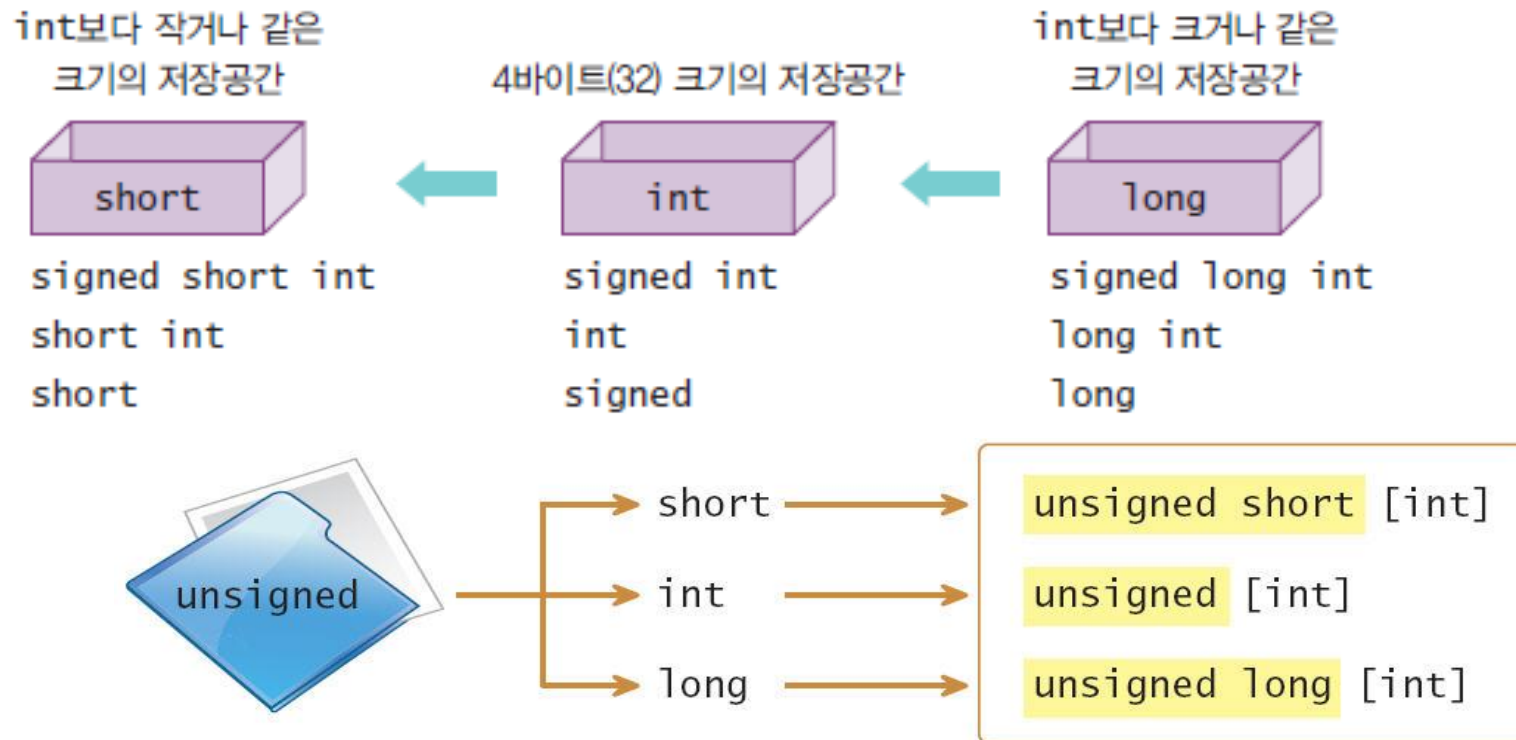
변수 선언문



3. 기본 자료형

◆ 정수 자료형

- ❖ short, int, long
- ❖ signed, unsigned



3. 기본 자료형

◆ 정수형 저장공간



표 3-1 정수 자료형의 표현 범위

음수지원 여부	자료형	크기	표현 범위
부호가 있는 정수형 signed	signed short	2 바이트	$-32,768(-2^{15}) \sim 32,767(2^{15}-1)$
	signed int	4 바이트	$-2,147,483,648(-2^{31}) \sim 2,147,483,647(2^{31}-1)$
	signed long	4 바이트	$-2,147,483,648(-2^{31}) \sim 2,147,483,647(2^{31}-1)$
부호가 없는 정수형 unsigned	unsigned short	2 바이트	$0 \sim 65,535(2^{16}-1)$
	unsigned int	4 바이트	$0 \sim 4,294,967,295(2^{32}-1)$
	unsigned long	4 바이트	$0 \sim 4,294,967,295(2^{32}-1)$

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 기본 자료형

◆ 정수형의 이용

실습예제 3-4
Prj04
04integer.c
정수 표현을 위한 C 언어의 다양한 자료형
난이도: ★

```

01  /* 소스: 04integer.c */
02
03  #include <stdio.h>
04
05  int main(void)
06  {
07      short sVar = 32000;    //-32767에서 32767까지
08      int iVar = -2140000000;  //약 21억 정도까지 저장 가능
09
10      printf("저장 값: %d %d\n", sVar, iVar);
11
12      //C99 이후 추가된 자료형: 64비트의 정수형 자원
13      long long dist1 = 2720000000000; //지구와 천왕성 간의 거리(km) 27억 2천
14      __int64 dist2 = 4500000000000;   //태양과 해왕성 간의 거리(km) 45억
15
16      printf("지구와 천왕성 간의 거리(km): %lld\n", dist1);
17      printf("태양과 해왕성 간의 거리(km): %lld\n", dist2);
18
19      return 0;
20  }

```

자료형 long long, __int64는 64비트로
약 922경까지 저장 가능

long long과 __int64를 출력할 경우,
형식제어문자를 %lld로 기술

결과
지구와 천왕성 간의 거리(km): 2720000000000
태양과 해왕성 간의 거리(km): 4500000000000

3. 기본 자료형

◆ 부동소수형 3가지

❖ float, double, long double

- float는 4바이트이며, double과 long double은 모두 8바이트

표 3-4 부동소수형의 표현범위

자료형	크기	정수의 유효자릿수	표현범위
float	4 바이트	6~7	1.175494351E-38F에서 3.402823466E+38F까지
double	8 바이트	15~16	2.2250738585072014E-308에서 1.7976931348623158E+308까지
long double	8 바이트	15~16	2.2250738585072014E-308에서 1.7976931348623158E+308까지

◆ 상수 표현

- float 상수
 - 숫자 뒤에 f나 F를 붙임

3. 기본 자료형

◆ 부동소수형 예제

실습예제 3-5	Prj05	05floatdouble.c	부동소수형 변수의 선언과 활용	난이도: ★
<pre>01 /* 소스: 05floatdouble.c */ 02 03 #include <stdio.h> 04 05 int main(void) 06 { 07 float x = 3.14F; //float x = 3.14;인 경우, 경고 발생 08 double y = -3.141592; //double 저장공간 크기는 float의 2배 09 long double z = 29.74; //double과 long double은 저장공간이 모두 64비트 10 11 printf("부동소수 값: %f %f %f\n", x, y, z); //모두 %f로 출력 가능 12 13 return 0; 14 }</pre>				
결과	저장 값: 3.140000 -3.141592 29.740000			

float 형 상수에는 3.14와 같이
반드시 F나 f를 붙이도록

3. 기본 자료형

◆ 문자 자료형

- ❖ char, signed char, unsigned char 세 가지 종류
 - 저장공간 크기는 모두 1바이트

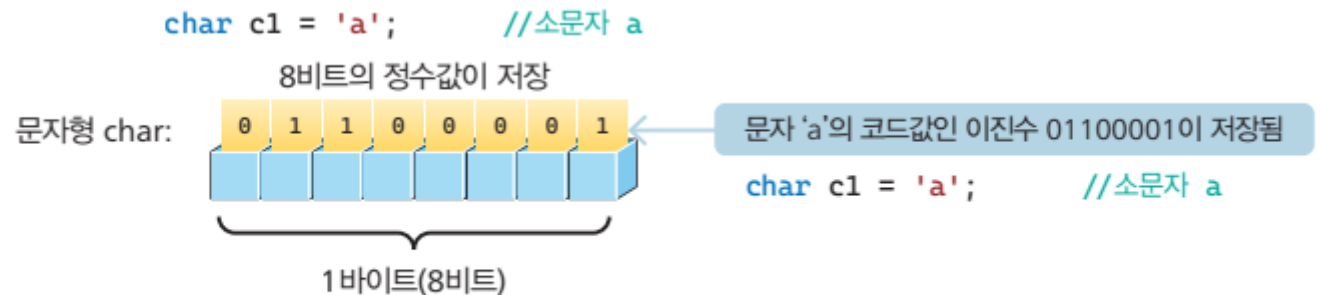


그림 3-36 문자형 자료 값의 표현과 저장공간

표 3-5 문자형의 표현범위

자료형	저장공간 크기	표현범위
char	1 바이트	-128에서 127까지(문자는 실제 0에서 127까지 이용)
signed char	1 바이트	-128에서 127까지
unsigned char	1 바이트	0 에서 255까지

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 기본 자료형

◆ 문자형 변수의 선언과 이용

실습예제 3-6	Prj06	06char.c	문자형 변수의 선언과 이용	난이도: ★
<pre> 01 /* 소스: 06char.c */ 02 03 #include <stdio.h> 04 05 int main(void) 06 { 07 char c1 = 'a'; //소문자 a 08 char c2 = 65; //대문자 A가 코드 값 65 09 char c3 = '\127'; //대문자 W의 8진수 코드 값 127 10 char c4 = '\x57'; //대문자 W의 16진수 코드 값 57 11 12 printf("문자 값(문자): %c %c %c %c\n", c1, c2, c3, c4); 13 printf("코드 값(번호): %d %d %d %d\n", c1, c2, c3, c4); 14 15 return 0; 16 } </pre>				
결과	<p>문자 값(문자): a A W W</p> <p>코드 값(번호): 97 65 87 87</p>			

3. 기본 자료형

◆ 자료형의 크기

표 3-7 기본 자료형의 저장공간 크기와 표현범위(자료형에서 []은 생략 가능함)

분류	자료형	크기	표현범위
문자형	char	1 바이트	+128(+2 ⁷) ~ 127(2 ⁷ +1)
	signed char	1 바이트	+128(+2 ⁷) ~ 127(2 ⁷ +1)
	unsigned char	1 바이트	0 ~ 255(2 ⁸ +1)
정수형	[signed] short [int]	2 바이트	+32,768(+2 ¹⁵) ~ 32,767(2 ¹⁵ +1)
	[signed] [int]	4 바이트	+2,147,483,648(+2 ³¹) ~ 2,147,483,647(2 ³¹ +1)
	[signed] long [int]	4 바이트	+2,147,483,648(+2 ³¹) ~ 2,147,483,647(2 ³¹ +1)
	[signed] long long [int]	8 바이트	9,223,372,036,854,775,808(+2 ⁶³) ~ 9,223,372,036,854,775,807(2 ⁶³ +1)
	unsigned short [int]	2 바이트	0 ~ 65,535(2 ¹⁶ +1)
	unsigned [int]	4 바이트	0 ~ 4,294,967,295(2 ³² +1)
	unsigned long [int]	4 바이트	0 ~ 4,294,967,295(2 ³² +1)
	[unsigned] long long [int]	8 바이트	0 ~ 18,446,744,073,709,551,615(2 ⁶⁴ +1)
부동소수형	float	4 바이트	대략 10 ⁺³⁸ ~ 10 ³⁸
	double	8 바이트	대략 10 ⁺³⁰⁸ ~ 10 ³⁰⁸
	long double	8 바이트	대략 10 ⁺³⁰⁸ ~ 10 ³⁰⁸

❖ 연산자 sizeof : 자료형, 변수, 상수의 저장공간 크기를 바이트 단위 반환

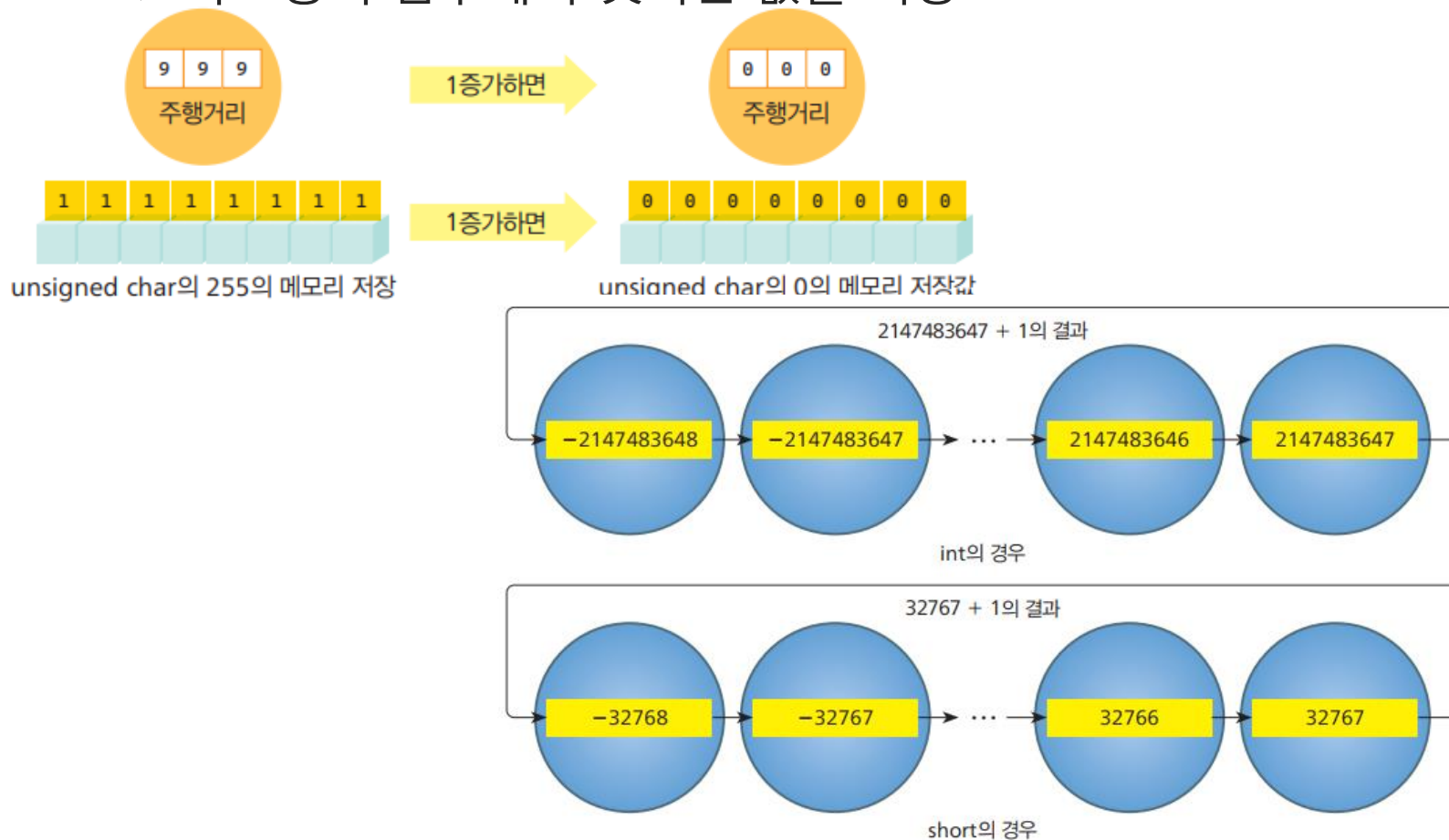
```
sizeof(char) // sizeof (자료형키워드), 괄호가 반드시 필요
sizeof 3.14  // sizeof 상수, sizeof (상수) 모두 가능
sizeof n     // sizeof 변수, sizeof (변수) 모두 가능
```

[출처] 강환수 외, Perfect C 3판, 인피니티북스

3. 기본 자료형

◆ 오버플로와 언더플로

❖ 자료형의 범주에서 벗어난 값을 저장



3. 기본 자료형

◆ 변수의 크기와 오버플로

실습예제 3-7	Prj07	07sizeflow.c	변수의 크기와 오버플로	난이도: ★★
	01	/* 소스: 07sizeflow.c */		
	02			
	03	#include <stdio.h>		
	04			
	05	int main(void)		
	06	{		
	07	printf(" 자료형 : 크기(바이트)\n");		
	08	printf(" char : %d\n", sizeof(char));		
	09	printf(" int : %d %d\n", sizeof(int), sizeof(200));		
	10	printf(" long long : %d %d\n", sizeof(long long), sizeof(900LL));		
	11	printf(" float : %d %d\n", sizeof(float), sizeof 3.14F);		
	12	printf("long double : %d %d\n", sizeof(long double), sizeof 3.24L);		
	13			
	14	short s = 32767;		
	15	printf("%d\n", s);		
	16	s = s + 1;	연산 32767 + 1의 결과는 32768이나 자료형 short에서 32768은 오버플로가 발생하므로 -32768이 저장됨	
	17	printf("%d\n", s); //오버플로 발생		
	18			
	19	return 0;		
	20	}		
결과			자료형 : 크기(바이트) char : 1 int : 4 4 long long : 8 8 float : 4 4 long double : 8 8 32767 -32768	

Ⅱ. 자료형과 변수선언

2교시 수업을 마치겠습니다.



The background of the slide features a dark blue color with a pattern of lighter blue geometric shapes, primarily triangles. On the left side, there is a vertical strip containing a composite image: a hand holding a bar chart at the top, and a laptop keyboard with glowing circular patterns at the bottom.

Ⅲ. 상수 표현 방법

1. 상수의 종류
2. 정수와 실수 리터럴 상수의 다양한 표현 방식
3. 열거형 상수와 매크로 상수

1. 상수의 종류

◆ 리터럴 상수와 심볼릭 상수

표 3-8 상수의 종류

구분	표현 방법	설명	예
리터럴 상수 (이름이 없는 상수)	정수형 실수형 문자 문자열 상수	다양한 상수를 있는 그대로 기술	32, 025, 0xf3, 10u, 100L, 30LL 3.2F, 3.15E3, 'A', '\n', '\0', '\24', '\x2f' "C 언어", "프로그래밍 언어\n"
심볼릭 상수 (이름이 있는 상수)	const 상수	키워드 const를 이용한 변수 선언과 같으며, 수정할 수 없는 변수 이름으로 상수 정의	const double PI = 3.141592;
	매크로 상수	전처리기 명령어 #define으로 다양한 형태를 정의	#define PI 3.141592
	열거형 상수	정수 상수 목록 정의	enum bool {FALSE, TRUE};

1. 상수의 종류

◆ 문자 상수

함수 printf()의 첫 인자인 문자열을 형식 제어문자열이라 하는데, %c와 같이 출력될 다른 값으로 대체되는 부분을 형식 제어문자라 한다. 형식 제어문자를 제외하고는 그대로 출력된다.

printf("%c ", 'A');

printf("%c", '\n');

A

(new line)

여기서 다음 출력 준비

문자 상수는 printf()에서 %c또는 %C로 출력

1. 상수의 종류

◆ 이스케이프 시퀀스

실습예제 3-8	Prj08	08charliteral.c	이스케이프 문자를 비롯해서 다양한 문자 리터럴의 표현	난이도: ★
<pre> 01 /* 소스: 08charliteral.c */ 02 03 #include <stdio.h> 04 05 int main(void) 06 { 07 printf("%Cava", 'J'); 08 09 char sq = '\\'; 10 printf("%c\\7\\n", '\\a'); 11 printf("%c자바 언어\\n", sq); 12 13 //문자열 내부에서는 "(큰따옴표) 반드시 \"로 사용 14 printf("\\nC언어\" 정말 재미있다!\\n"); 15 16 return 0; 17 } </pre>				
결과	<p>Java 출력</p> <p>경고음 소리가 출력되며, 뒤 이은 \7도 \a와 같으므로 경고음이 2번 울린다.</p> <p>//작은따옴표</p> <p>//알람 문자를 2번 출력하고 공백 줄</p> <p>//문자열 내부에서는 '(작은따옴표) 그대로 사용 가능</p> <p>//문자열 내부에서는 "(큰따옴표) 반드시 \"로 사용</p> <p>Java</p> <p>'자바 언어'</p> <p>"C언어" 정말 재미있다!</p>			

1. 상수의 종류

◆ 이스케이프 시퀀스

표 3-9 이스케이프 문자

제어문자 이름	영문 표현	코드값(10진수)	\ddd(8진수)	제어문자 표현	의미
널문자	NULL	0	\000	\0	아스키코드 0번
경고	BEL(Bell)	7	\007	\a	경고음이 울림
백스페이스	BS(Back Space)	8	\010	\b	커서를 한 문자 뒤로 이동
수평탭	HT(Horizontal Tab)	9	\011	\t	커서를 수평으로 다음 탭만큼 이동
개행문자	LF(Line Feed)	10	\012	\n	커서를 다음 줄로 이동
수직탭	VT(Vertical Tab)	11	\013	\v	수직으로 이동하여 탭만큼 이동
폼피드	FF(Form Feed)	12	\014	\f	새 페이지의 처음으로 이동
캐리지 리턴	CR(Carriage Return)	13	\015	\r	커서를 현재 줄의 처음으로 이동
큰따옴표	Double quote	34	\042	\"	" 문자
작은따옴표	Single quote	39	\047	\'	' 문자
역슬래쉬	Backslash	92	\134	\\	\ 문자

2. 정수와 실수 리터럴 상수의 다양한 표현 방식

표 3-11 리터럴 상수 접미어

구분	표현 방법	접미어	예
정수형 상수	unsinged (int)	u	1000000u
		U	24563U
	unsigned long	ul	87252987ul
		UL	10000000UL
	unsigned long long	ull	1000000000000000ull
		ULL	2450000000000000ULL
실수형 상수	float	l	-20987l
		L	76528876L
	long long	ll	1000000000000000ll
		LL	-2450000000000000LL
	long double	l	4356.9876l
		L	5634.984276L

2. 정수와 실수 리터럴 상수의 다양한 표현 방식

실습예제 3-9	Prj09	09numliteral.c	정수형과 실수형 리터럴 상수의 다양한 표현	난이도: ★
	<pre> 01 /* 소스: 09numliteral.c */ 02 03 #include <stdio.h> 04 05 int main(void) 06 { 07 printf("%d, %d\n", 010, 015); //8진수 08 printf("%d, %d\n", 10, 15); //10진수 09 printf("%d, %d\n", 0X1a, 0x15); //16진수 10 11 printf("%f, ", 2.71828); 12 printf("%f, ", 2.71828E+2); 13 printf("%f\n", 2.71828e-2); 14 15 return 0; 16 } </pre> <div data-bbox="993 935 1350 1013"> <p>부동소수의 10의 지수승 표현으로 부동소수는 printf()에서 %f로 출력</p> </div>			
결과	<pre> 8, 13 10, 15 26, 21 2.718280, 271.828000, 0.027183 </pre>			

2. 정수와 실수 리터럴 상수의 다양한 표현 방식

◆ 키워드 const

❖ 초기 값을 수정할 수 없는 이름이 있는 심볼릭 상수

실습예제 3-10	Prj10	10const.c	키워드 const를 사용한 상수 선언	난이도: ★
<pre> 01 /* 소스: 10const.c */ 02 03 #include <stdio.h> 04 05 int main(void) 06 { 07 //키워드 const로 상수 만들기 08 double const e = 2.718281; //오일러 수 09 10 //e = 2.71828; 11 printf("오일러 수 %f\n", e); 12 13 return 0; 14 } </pre>				
<div style="display: flex; align-items: center;"> <div style="border: 1px solid red; padding: 5px; margin-left: 20px;"> 상수는 수정할 수 없으므로 주석을 빼면 컴파일 오류가 발생 </div> </div>				
결과	오일러 수 2.718281			

3. 열거형 상수

◆ 키워드 enum

❖ 정수형 상수 목록 집합을 정의하는 자료형

- 목록 첫 상수의 기본 값이 0이며
 - 다음부터 1씩 증가하는 방식으로 상수 값이 자동으로 부여
- 상수 값을 지정한 상수는 그 값으로
- 따로 지정되지 않은 첫 번째 상수는 0이며,
 - 중간 상수는 앞의 상수보다 1씩 증가한 상수 값으로 정의

```
enum SHAPE { POINT, LINE, TRI = 3, RECT, OCTA = 8, CIRCLE };
```

정수형 상수 목록

POINT	0	LINE	1	TRI	3	RECT	4	OCTA	8	CIRCLE	9
-------	---	------	---	-----	---	------	---	------	---	--------	---

```
enum boolean {FALSE, TRUE};
```

```
enum city {SEOUL, INCHEON, DAEGU, PUSAN};
```

```
enum OS {WINDOW, OSX = 3, ANDROID, IOS = 7, LINUX};
```

```
enum pl {c = 1972, cpp = 1983, java = 1995, csharp = 2000};
```

3. 열거형 상수

실습예제 3-11	Prj11	11enum.c	enum의 열거형 상수	난이도: ★★
<pre> 15 /* 소스: 11enum.c */ 16 17 #include <stdio.h> 18 19 int main(void) 20 { 21 // 키워드 enum으로 열거형 정수 상수 목록 만들기 22 enum DAY { SUN, MON, TUE, WED, THU, FRI, SAT }; 23 printf("%d %d\n", SUN, THU); // 0 4 24 25 // 상수 목록에서 특정한 정수 지정 가능 26 enum SHAPE { POINT, LINE, TRI = 3, RECT, OCTA = 8, CIRCLE }; 27 printf("LINE: %d, RECT: %d, CIRCLE: %d\n", LINE, RECT, CIRCLE); 28 29 enum pl { c = 1972, cpp = 1983, java = 1995, csharp = 2000 }; 30 printf("c: %d, cpp: %d, java: %d\n", c, cpp, java); 31 32 return 0; 33 } </pre>				
<div style="border: 1px solid red; padding: 5px; width: fit-content; margin: 10px auto;"> 상수 SUN은 0에서부터 순차적으로 1씩 증가되어 지정되며, 상수 THU는 4로, SAT는 6으로 지정 </div>				
결과	<pre> 0 4 LINE: 1, RECT: 4, CIRCLE: 9 c: 1972, cpp: 1983, java: 1995 </pre>			

Ⅲ. 상수 표현 방법

3교시 수업을 마치겠습니다.

