

CHAPTER 02

기본

01 기본 용어

02 출력

03 기본 자료형

04 변수

05 복합 대입 연산자

06 증감 연산자

07 자료형 검사

08 var 키워드

09 입력

10 자료형 변환

요약

연습문제

학습 목표

- C#에서 사용하는 기본 용어를 이해한다.
- 기본 출력 방법을 익힌다.
- 기본 자료형과 변수, 이와 관련된 연산자의 사용법을 익힌다.
- 자료형 변환 방법을 익힌다.

01 다음 빈칸을 채우시오.

- ① 표현식이 하나 이상 모이고, 마지막에 세미콜론이 찍히면 (**문장**)이 된다.
- ② (**키워드**)는 특별한 의미가 부여된 단어로 C#이 처음 만들어질 때 정해졌다.
- ③ (**식별자**)는 이름을 붙일 때에 사용하는 단어이다.
- ④ (**주석**)은 프로그램의 진행에 전혀 영향을 주지 않는 코드로 프로그램을 설명하는 데 사용하는 코드이다.
- ⑤ 문자의 집합을 (**문자열**)이라고 부른다.
- ⑥ 코드를 실행하는 중에 발생하는 오류를 (**예외**) 또는 **런타임 에러**라고 부른다.
- ⑦ (**불**)은 참과 거짓을 표현할 때에 사용한다.
- ⑧ 두 피연산자의 크기 등을 비교해서 불 자료형을 만드는 연산자를 **비교 연산자**라고 부른다.
- ⑨ (**변수**)는 값을 저장할 때 사용하는 식별자이다.
- ⑩ (**var**) 키워드로 변수를 만들 때 자료형을 자동으로 지정할 수 있다.

02 다음 문장이 맞다면 O, 틀리다면 X 하시오.

- ① 논리 연산자 !, &&, ||는 모두 이항 연산자이다. **X: 논리 부정 연산자(!)는 단항 연산자입니다.**
- ② sizeof 연산자는 int, long, char, string, float, double, bool 자료형 등의 크기를 구할 수 있다. **X: string 자료형을 구할 수 없습니다.**
- ③ var 키워드로 선언한 변수는 자료형의 제약으로부터 자유로우므로, 여러 자료형의 값을 할당할 수 있다. **X: 한 번 결정된 자료형을 변경할 수는 없습니다.**
- ④ var 키워드는 선언과 동시에 값을 할당해야 한다. **O**
- ⑤ Console.ReadLine() 메서드의 결과는 무조건 string 자료형이다. **O**
- ⑥ 문자열을 다른 자료형으로 변환할 때는 toString() 메서드를 사용한다. **X: 대소문자를 구분해서 ToString() 메서드입니다.**
- ⑦ double 자료형 또는 float 자료형의 숫자를 int 자료형으로 변환하면, 소수점 아래의 정보를 최대한 유지하기 위해 반올림해서 정수로 변환한다. **X: 소수점을 그냥 잘라서 내림합니다.**
- ⑧ int 자료형의 숫자 -2147483648는 양수 값이 없는 숫자이므로, -(2147483648)로 사용해도 -2147483648가 나온다. **O**

03 다음 중에서 숫자 관련 자료형이 아닌 것은?

- ① string ② int ③ double ④ float ⑤ long

04 다음 중 변수 pi의 자료형이 double 자료형으로 선언되는 것을 고르시오.

- ① var a = 3.141592;
 ② var b = 3.141592f;
 ③ var c = 3.141592l;
 ④ var d = 3.141592i;

05 다음 중에서 이스케이프 문자 중에 개행(줄 바꿈)에 사용되는 것은 무엇인가?

- ① \e ② \t ③ \n ④ \\

06 다음 중에서 sizeof 연산자를 적용할 수 없는 자료형은?

- ① string ② int ③ char ④ float ⑤ bool

07 'a' + 'b'의 실행 결과로 옳은 것을 고르시오.

- ① 문자 'ab' ② 문자열 "ab" ③ 정수 195 ④ 문자열 "Ã"

08 알맞은 자료형을 빈칸에 쓰세요.

- ① [**string**] name = "대한민국"
- ② [**long**] sum = 41474836491;
- ③ [**float**] pi_a = 3.14f;
- ④ [**long**] pi_b = 3.14

09 연산자 =와 ==의 차이점을 설명하시오.

= 연산자는 변수에 값을 할당하는 연산자이고, == 연산자는 양쪽 변의 대상이 같은지를 확인하는 연산자이다.

10 var 키워드를 사용할 수 있는 기본 조건을 설명하시오.

지역 변수로 선언하는 경우 / 변수를 선언과 동시에 초기화하는 경우

11 inch 단위를 입력 받아 cm 단위를 구하는 코드를 작성하시오.

HINT 1inch = 2.54cm

```
Console.Write("inch 단위 입력: ");
double inch = double.Parse(Console.ReadLine());
Console.WriteLine(inch + "inch는 " + (inch * 2.54) + "cm입니다.");
```

12 킬로그램(kg) 단위를 입력 받아 파운드(pound) 단위를 구하는 코드를 작성하시오.

HINT 1kg = 2.20462262pound

```
Console.Write("kg 단위 입력: ");
double kg = double.Parse(Console.ReadLine());
Console.WriteLine(kg + "kg은 " + (kg * 2.20462262) + "pound입니다.");
```

13 원의 반지름을 입력 받아 원의 둘레와 넓이를 구하는 코드를 작성하시오.

HINT 둘레 = 2 * PI * 반지름, 넓이 = PI * 반지름 * 반지름

```
Console.Write("원의 반지름 입력: ");
double radius = double.Parse(Console.ReadLine());
Console.WriteLine("원의 둘레: " + (2 * 3.14 * radius));
Console.WriteLine("원의 넓이: " + (3.14 * radius * radius));
```

14 다음 코드의 실행 결과를 예측하시오.

```
static void Main(string[] args)
{
    Console.WriteLine(52 + 273);
    Console.WriteLine(52 + "273");
    Console.WriteLine("52" + 273);
    Console.WriteLine("52" + "273");
}
```

325
52273
52273
52273

CHAPTER 03

조건문

01 if 조건문

02 if else 조건문

03 중첩 조건문

04 if else if 조건문

05 switch 조건문

06 삼항 연산자

07 함께하는 응용예제

요약

연습문제

학습 목표

- if 조건문의 기본적인 사용 방법을 익힌다.
- if 조건문과 논리 연산자를 함께 사용하는 방법을 익힌다.
- switch 조건문을 이해한다.

01 다음 괄호를 채우시오.

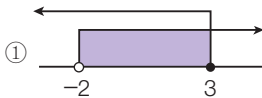
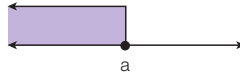
- ① if 조건문은 가장 기본적인 조건문이다. 불_표현식이 (**true**)면 문장을 실행하고 (**false**)면 문장을 무시한다.
- ② switch 조건문의 괄호에 입력한 표현식이 case 키워드 뒤에 있는 표현식과 하나도 맞지 않는 경우, (**default**) 키워드 뒤의 문장을 실행한다.
- ③ 문자열이 어떤 문자열을 포함하고 있는지 확인할 때는 (**Contains()**) 메서드를 사용한다.

02 다음 문장이 맞다면 O, 틀리다면 X 하시오.

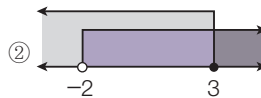
- ① if 조건문에서 실행할 문장들이 여러 줄이라면 반드시 중괄호를 사용해서 묶어줘야 한다. **O**
- ② switch 조건문에서 case 키워드 뒤에 오는 문장들이 여러 줄이라면 반드시 중괄호를 사용해서 묶어줘야 한다. **X: case 키워드 뒤에 오는 문장은 따로 중괄호로 묶지 않아도 괜찮습니다 (일반적으로 안 묶습니다).**
- ③ switch 구문에서 case 키워드 뒤에 오는 문장들은 반드시 마지막이 break 키워드로 끝나야 한다. 그렇지 않으면 오류가 발생한다.
X: 일반적으로 그렇게 사용하기는 하지만, 오류가 발생하지는 않습니다.

03 다음과 같은 범위 표현이 있을 때, C#의 비교 연산자, 조건 연산자, 변수 x를 활용해 색칠한 부분의 범위를 표현하시오(검은 점은 포함, 흰 점은 포함하지 않는다는 것을 의미).

HINT $x \leq a$ 를 그림으로 표현한 예시



$-2 < x \ \&\& \ x \leq 3$



$5 \leq x \ \&\& \ x < 7$



$x \leq -20 \ || \ 20 \leq x$

04 다음 중에서 조건문과 직접적으로 관련된 키워드가 아닌 것은 무엇인가?

- ① if ② else ③ **var** ④ switch

05 다음 중에서 switch 조건문과 직접적으로 관련된 키워드가 아닌 것은 무엇인가?

- ① switch ② else ③ case ④ default

06 다음 중 조건부 표현식의 기본적인 형태로 알맞은 것을 고르시오.

- ① 불 표현식 ? 참일 때 : 거짓일 때 ② if 불 표현식 then 참일 때 else 거짓일 때
 ③ 참일 때 if 불 표현식 else 거짓일 때 ④ 불 표현식 and 참일 때 or 거짓일 때

07 사용자에게 태어난 년도를 입력 받아 그 해의 띠를 출력하는 프로그램을 작성하시오.

HINT 입력 받은 년도를 12로 나눈 나머지를 사용한다. 나머지가 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11일 때에 각각 원숭이, 닭, 개, 돼지, 쥐, 소, 범, 토끼, 용, 뱀, 말, 양띠이다.

```
Console.WriteLine("태어난 년도: ");
int birth = int.Parse(Console.ReadLine());
switch (birth % 12)
{
    case 0: Console.WriteLine("원숭이 띠입니다."); break;
    case 1: Console.WriteLine("닭 띠입니다."); break;
    case 2: Console.WriteLine("개 띠입니다."); break;
    case 3: Console.WriteLine("돼지 띠입니다."); break;
    case 4: Console.WriteLine("쥐 띠입니다."); break;
    case 5: Console.WriteLine("소 띠입니다."); break;
    case 6: Console.WriteLine("호랑이 띠입니다."); break;
    case 7: Console.WriteLine("토끼 띠입니다."); break;
    case 8: Console.WriteLine("용 띠입니다."); break;
    case 9: Console.WriteLine("뱀 띠입니다."); break;
    case 10: Console.WriteLine("말 띠입니다."); break;
    case 11: Console.WriteLine("양 띠입니다."); break;
}
```


08 사용자에게 현재 월을 입력 받아 계절을 출력하는 프로그램을 작성하시오.

```

Console.WriteLine("현재가 몇 월인지 입력해주세요: ");
int year = int.Parse(Console.ReadLine());
if (3 <= year && year <= 5)
{
    Console.WriteLine("봄입니다.");
}
else if (6 <= year && year <= 8)
{
    Console.WriteLine("여름입니다.");
}
else if (9 <= year && year <= 11)
{
    Console.WriteLine("가을입니다.");
}
else
{
    Console.WriteLine("겨울입니다.");
}
    
```

09 다음 중첩 조건문을 논리 연산자로 하나의 if 조건문으로 작성하시오.

```

if (x > 10)
{
    if (x < 20)
    {
        Console.WriteLine("조건에 맞습니다.");
    }
}
    
```

```

if (x > 10 && x < 20)
{
    Console.WriteLine("조건에 맞습니다.");
}
    
```

10 다음 코드의 주석 부분이 입력 값일 때에 실행 결과를 쓰시오.

```
static void Main(string[] args)
{
    int x, y;
    /* 주석 부분 */

    if (x > 4)
    {
        if (y > 2)
        {
            Console.WriteLine(x * y);
        }
    }
    else
    {
        Console.WriteLine();
    }
}
```

모두 개행(\n) 하나 출력한다.

11 다음 if 조건문으로 작성된 프로그램을 switch 조건문으로 옮기시오.

```
static void Main(string[] args)
{
    Console.Write("학년을 입력하세요: ");
    int level = int.Parse(Console.ReadLine());

    if (level == 1)
    {
        Console.WriteLine("수강해야 하는 전공 학점: 12학점");
    }
    else if (level == 2)
    {
        Console.WriteLine("수강해야 하는 전공 학점: 18학점");
    }
    else if (level == 3)
    {
        Console.WriteLine("수강해야 하는 전공 학점: 10학점");
    }
    else if (level == 4)
    {
        Console.WriteLine("수강해야 하는 전공 학점: 18학점");
    }
}
```

```
Console.Write("학년을 입력하세요");
int level = int.Parse(Console.ReadLine());
switch(level)
{
    case 1:
        Console.WriteLine("수강해야 하는 전공 학점: 12학점");
        break;
    case 2:
        Console.WriteLine("수강해야 하는 전공 학점: 18학점");
        break;
    case 3:
        Console.WriteLine("수강해야 하는 전공 학점: 10학점");
        break;
    case 4:
        Console.WriteLine("수강해야 하는 전공 학점: 18학점");
        break;
}
```

CHAPTER 04

반복문

- 01 반복문과 배열
- 02 while 반복문
- 03 do while 반복문
- 04 for 반복문
- 05 역 for 반복문
- 06 foreach 반복문
- 07 중첩 반복문
- 08 break 키워드
- 09 continue 키워드
- 10 함께하는 응용예제

요약

연습문제

학습 목표

- 배열을 생성하는 방법과 사용하는 방법을 익힌다.
- while과 do while 반복문을 이해한다.
- for와 foreach 반복문을 이해한다.
- break와 continue 키워드를 이해한다.

01 다음 괄호를 채우시오.

- ① 배열 안에 들어있는 각각의 자료를 (요소)라고 부른다.
- ② 배열의 요소에 접근할 때 대괄호 안에 넣는 숫자를 (인덱스)라고 부른다.
- ③ 배열의 요소 개수가 몇 개 있는지 확인할 때는 (Length) 속성을 사용한다.
- ④ (break) 키워드는 switch 조건문 또는 반복문을 벗어날 때 사용하는 키워드이다.
- ⑤ (continue) 키워드는 반복문 내부에서 현재 반복을 멈추고 다음 반복을 진행하게 만드는 키워드이다.
- ⑥ 자신을 변경하지 않고 반환(리턴)만 하는 메서드를 비파괴적 메서드라고 부른다.

02 다음 중 배열의 길이를 넘는 요소에 접근할 때 발생하는 오류를 고르시오.

- ① ArrayOutOfRangeException
- ② IndexOutOfRangeException
- ③ CountOutOfRangeException
- ④ ListOutOfRangeException

03 다음 중 반복문과 관련된 설명으로 틀린 것을 고르시오.

- ① 무한하게 반복하는 반복문을 무한 반복문이라고 부른다.
- ② while 반복문은 조건을 나중에 검사하고 코드 블록을 반복해서 실행하며, do while 반복문은 조건을 먼저 검사하고 코드 블록을 반복해서 실행한다.
- ③ for 반복문은 일반적으로 특정한 횟수만큼 반복을 돌릴 때 사용한다.
- ④ do while 반복문은 일반적으로 문장을 적어도 한 번 꼭 실행해야 하는 특수한 목적에 사용한다.

04 다음 중에 배열을 생성하는 코드로 잘못된 것은?

- ① `int[] array = new int[20]`
- ② `int[] array = new int[3] { 1, 2, 3 }`
- ③ `int[] array = int[] { 1, 2, 3 }`
- ④ `int[] array = {1, 2, 3}`

05 다음 중 문자열 양 옆의 공백을 제거할 때 사용하는 함수를 고르시오.

- ① `ToUpper()`
- ② `Trim()`
- ③ `Strip()`
- ④ `Merge()`

06 다음 중 배열의 요소를 결합해서 문자열로 만드는 함수를 고르시오.

- ① `Merge()`
- ② `Join()`
- ③ `Connect()`
- ④ `Strip()`

07 다음 중 `foreach` 반복문의 기본적인 형태를 고르시오.

- ① `foreach (var item : collection) {}`
- ② `foreach (var item in collection) {}`
- ③ `for (var item : collection) {}`
- ④ `for (var item in collection) {}`

08 다음 코드의 실행 결과를 예측해보시오.

```
static void Main(string[] args)
{
    // 변수를 선언합니다.
    int[] array = new int[10];

    // 반복을 수행합니다.
    foreach (var item in array)
    {
        // 출력합니다.
        Console.WriteLine(item);
    }
}
```

0을 10개 출력한다.

09 다음 코드의 실행 결과를 예측해보시오.

```
static void Main(string[] args)
{
    for (int i = 0; i < 20; i = i + 2)
    {
        Console.Write("출력");
    }
}
```

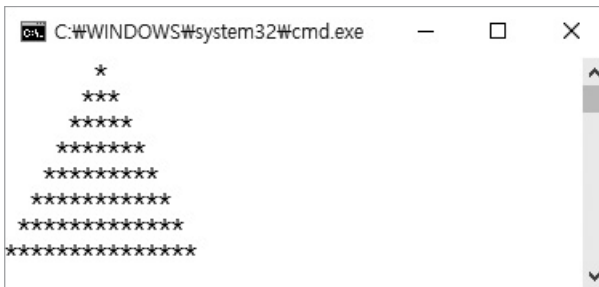
출력을 10번 출력한다.

10 다음 코드의 for 반복문을 while 반복문으로 바꿔보시오.

```
static void Main(string[] args)
{
    for (int i = 0; i < 10; i++)
    {
        Console.WriteLine("출력");
    }
}
```

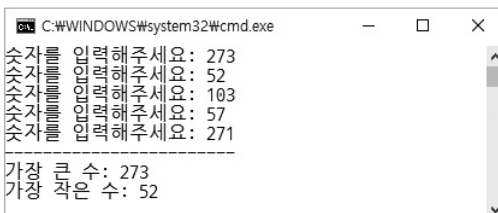
```
int i = 0;
while (i < 10)
{
    Console.WriteLine("출력");
    i++;
}
```

11 다음 출력을 생성하는 코드를 만들어보시오.



```
for (int i = 0; i < 8; i++)
{
    for (int j = 7; j >= i; j--)
    {
        Console.Write(" ");
    }
    for (int j = 0; j < 2 * i + 1; j++)
    {
        Console.Write("*");
    }
    Console.WriteLine();
}
```

12 다음과 같이 5개의 수를 입력 받아, 가장 작은 수와 가장 큰 수를 출력하는 프로그램을 작성하시오.



```
int max = int.MinValue;
int min = int.MaxValue;
for (int i = 0; i < 5; i++)
{
    Console.WriteLine("숫자를 입력해주세요: ");
    int input = int.Parse(Console.ReadLine());
    if (input < min) { min = input; }
    if (input > max) { max = input; }
}
```


13 다음은 읽고 말하기 수열(또는 개미 수열이라고도 부르는)의 규칙입니다.

1, 11, 12, 1121, 122111, 112213, ...

첫 번째 수열: 1

두 번째 수열: 1이 1개 = 11

세 번째 수열: 1이 2개 = 12

네 번째 수열: 1이 1개, 2가 1개 = 1121

다섯 번째 수열: 1이 2개, 2가 1개, 1이 1개 = 122111

여섯 번째 수열: 1이 1개, 2가 2개, 1이 3개 = 112213

이러한 수열의 20번째 숫자를 프로그램을 만들어 찾으시오.

```
// 첫 번째 수열 입력
string start = "1";
// 계산 시작
for (int i = 0; i < 20; i++)
{
    // 출력
    Console.WriteLine((i + 1) + "번째: " + start);
    // 읽고 말하기 수열 계산
    string end = "";
    char number = start[0];
    int count = 0;
    for (int j = 0; j < start.Length; j++)
    {
        if (number != start[j])
        {
            end = end + number + count;
            number = start[j];
            count = 1;
        }
        else
        {
            count++;
        }
    }
    end = end + number + count;
    // 다음 단계 준비
    start = end;
    페이지 6 / 18
}
```