﻿변수, 형변환, 연산자

﻿

**변수(variable)**

**변수**

프로그램 작업을 처리하기 위해 하나의 값을 저장할 수 있는 메모리공간

임의의 메모리 공간에 이름을 붙여 관리함

자바에서는 다양한 타입을 저장할 수 없고, 한가지 타입만 값으로 저장될 수 있음

|  |  |  |  |
| --- | --- | --- | --- |
| 기본 자료형 | | | |
| 자료형 | 키워드 | 크기 | 표현범위 |
| 논리형 | boolean | 1 byte | true, false |
| 문자형 | char | 2 byte | 0~65,535 |
| 정수형 | byte | 1 byte | -128 ~ 127 (-27~ 27-1) |
| short | 2 byte | -32,768 ~ 32,767 (-215~ 215-1) |
| int | 4 byte | -2,147,483,648 ~ 2,147,483,647 |
| long | 8 byte | -9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807 |
| 실수형 | float | 4 byte | -3.4E38 ~ +3.4E38 |
| double | 8 byte | 1.7E308 ~ + 1.7E308 |

* 셀 병합
* 행 분할
* 열 분할
* 너비 맞춤
* 삭제

**변수의 선언 방법**

**1. 자료형 변수명;**

변수의 선언과 값의 대입을 분리하는 방법

int num1;

num1 = 10;

**2. 자료형 변수명 = 값;**

변수의 선언과 동시에 값을 대입하는 방법

int num2 = 20;

**3. 자료형 변수명1, 변수명2;**

동일한 자료형 타입의 변수를 다수개 선언하는 방법

int num3, num4

**4. 자료형 변수명1 = 값1, 변수명2 = 값2;**

동일한 자료형 타입의 변수를 다수개 선언하면서 값을 대입하는 방법

**기본 데이터 type 참조 데이터 type**

**1.기본 데이터 type(=기본 자료형)**

메모리에 있는 실제 값 = 변수 데이터 값

ex) int, double

**2. 참조 데이터(객체 자료형)**

메모리에 있는 실제 값 = 변수가 저장된 주소

ex) String

**형변환**

**1. 묵시적 형변환**

작은 주머니의 데이터를 큰 주머니로 옮기는 것

**2. 명시적 형변환**

큰 주머니의 데이터를 작은 주머니로 옮기는 것(데이터 손실이 있을 수 있음)

package com.lec.ex;

public class VarEx06 {

public static void main(String[] args) {

int i1 = 10; // 4byte

long l = 2200000000l; // 8byte

double d1 = 10; // 8bye = 4byte 묵시적 형변환 (작은 byte값을 큰 byte공간에)

System.out.println("d1="+d1);

double d2 = 10.19;

int i2 = (int)d2; // 명시적인 형변환(데이터 손실이 발생할 수도 있음)

System.out.println("i2="+i2);

}

}



**연산자 (operator)**

일정한 규칙을 가지고 계산하는 것

우리가 일상생활에서 사용하고 있는 덧셈, 뺄셈, 곱셉, 나눗셈과

자동증감, 동등비교, 비트연산 등의 많은 연산자가 있음

셀 전체 선택

열 너비 조절

행 높이 조절

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 우선순위 | 명칭 | | 연산자 | 결합성 |
| 1 | 1차 연산자 | | ( ) | → |
| 2 | 단항 연산자 | | ! ++ -- | ← |
| 3 | 이  항  연  산  자 | 승법연산자 | \* / % | → |
| 4 | 가법연산자 | + - |
| 5 | 관계(비교)연산자 | < <= > >= == != |
| 6 | 비트곱연산자 | & |
| 7 | 비트합연산자 | | |
| 8 | 논리곱연산자 | && |
| 9 | 논리합연산자 | || |
| 10 | 조건(3항)연산자 | | ?: |
| 11 | 할당연산자 | | = += -= \*= /= %= | ← |

**1. 산술연산자**

+ : 더하기

- : 빼기

\* : 곱하기

/ : 나누기

% : 나머지

**2. 증감연산자**

++ : 1을 더한다

-- : 1을 뺀다

위치에 따라서 다른 결과가 나올 수 있음

int n1 = 10;

int n2 = ++n1; // n1 값이 하나 증가

System.out.println("n1="+n1+"\t n2="+n2);

//n1=11 n2=11가 출력

int n3 = 10;

int n4 = n3--; // ;이후에 n3 값이 하나 감소

System.out.println("n3="+n3+"\t n4="+n4);

//n3=9 n4=10가 출력

int n5 = 10;

++n5; // n5 값이 하나 증가

int n6 = n5; // 증가 된 n5 값을 n6에

System.out.println("n5="+n5+"\t n6="+n6);

//n5=11 n6=11가 출력

int n7 = 10;

int n8 = n7; // n7의 값을 n8에 대응

++n7; // n7 값이 하나 증가

System.out.println("n7="+ n7+"\t n8="+ n8);

//n7=11 n8=10가 출력



**3. 비교연산자/관계연산자**

>

>=

<

<=

== : 같다

!= : 다르다

**4. 논리연산자**

&& : AND

&&연산자의 좌항이 false일 경우 우항 실행 안함

|| : OR

||연산자의 좌항이 true일 경우 우항 실행 안함

**5. 삼항연산자**

(조건) ? (true일 경우 출력할 값/식) : (false일 경우 출력할 값/식)

**6. 대입(할당)연산자**

=

+=

-=

\*=

/=

%=

변형된 타입의 대입연산자들

자지 자신에 연산한 후의 결과값을 대입

int n1 = 10;

n1 += 10; // n1 변수의 값을 10 증가

n1 \*= 3; // n1 변수의 값에 3을 곱함

int n2;

int n1 = n2 = 10;



**07.비트연산자**

잘 사용하지 않음

비트를 사용해 연산하는 연산자

&

|

^

int i = 3; // 0 0 1 1

int j = 1; // 0 0 0 1

//-----------

// 0 0 0 1

System.out.println(i & j); // 1 이 출력

int i = 3; // 0 0 1 1

int j = 1; // 0 0 0 1

//-----------

// 0 0 1 1

System.out.println(i | j); // 3 이 출력

int i = 3; // 0 0 1 1

int j = 1; // 0 0 0 1

//-----------

// 0 0 1 0

System.out.println(i ^ j); // 2 가 출력



출력 관련 코드 정리

System.out.print(); // 개행을 미포함한 출력

System.out.println(); // 개행을 포함한 출력

System.out.printf(); // 특정 포맷으로 출력(개행 미포함)

char c1 = 'A';

char c2 = 'B';

System.out.printf("c1=%c \t c2=%c\n", c1, c2);

//%c 문자, %d 정수, %f 실수, %s 문자열

int i = 10;

byte j = 10;

double h = 10.15;

System.out.printf("i=%d, j=%d, h=%4.1f \n", i, j, h);

//%4.1f=전체 4자리 중에 소수점 한자리수까지 출력, 알아서 반올림



사용자가 값을 입력할 수 있게 해주는 코드

package com.lec.quiz;

import java.util.Scanner;

public class Quiz1 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("수를 입력하세요 : ");

int su = scanner.nextInt(); // 사용자로부터 정수 입력 받기

String result = su%3 == 0 ? "입력하신 수는 3의 배수" :"입력하신 수는 3의 배수가 아님";

System.out.println(result);

scanner.close();

}

}



﻿