

7월 15일 자바

//javac Helloworld.java -> Helloworld.class 파일이 생성된다.

//dir

//java Helloworld

//public static void main(String args[]) {} -> 이게 실행 가능한 메서드이기 때문에

//메인 메서드 없이 cmd창에서 java Helloworld로 실행하려고 하면 오류가 나고 오류 메시지는  
메인메서드를 찾을 수 없다고 뜬다.

//main 이 들어가야 실행이 된다.

항상 실행하기 전에 컴파일을 해주어야 한다.

순서! 저장 -> 컴파일->실행

////////////////////////////////

C:\JavaWork>javac Test001.java

C:\JavaWork>java Test001

////////////////////////////////

//변수는 값을 저장하는 기억공간

자바 자료형 8개 byte short long int (정수값 저장용 변수)

float boolean double char

////////////////////////////////

Test001

int i; //변수선언 자료형 변수이름; => 자료형만큼의 기억공간을 확보한다는 뜻이다. 기억공간을 확보하는 이유는 값을 저장하기 위해서 이다

//변수 선언시 두개의 자료형은 사용할 수 없다.

i = 100; // 대입 - 만들어진 공간에 100 저장.

i = 200; // 변수의 값은 바뀔수 있고 맨 마지막 값만 기억하게 된다.

System.out.println(i);

}

}

//자바 자료형 8개 byte(8) short(16) long(64) int(32) // float boolean double char

//정수값 저장용 변수 byte short int long.. // 기억공간의 크기는 제각각이기 때문에 다른 자료형을 써준다.

//변수는 값을 저장하는 기억공간

Test003

byte i;

i= 100;

i=200;

-->컴파일 시 에러.. i 공간이 200보다 작기 때문.

byte i;

i=100;

i=20;

--->에러 없음

Test004

j=i; // (오른쪽변수)i의 값이 복사되어 (왼쪽 변수)j에 대입이 되는 개념..

Test005

i = i+1;

// i+1은 변수i 가 가진 값과 연산을 하는 것

// 변수와 값은 연산이 가능하다. 이때는 변수가 가진 값과 값이 연산된다고 보면 된다.

//대입시에는 오른쪽이 먼저 동작하고 다음에 대입된다.

Test007

double j = 10.0; //j는 실수를 저장할 수 있는 기억공간.

System.out.println(j);

System.out.println(j+10); //출력값 20.0

/\*10.0+10

10.0+10.0 => 자동 변환이 된 것이다.

반면 10.0에서 10으로는 자동 변환이 되지 않는다.

System.out.println((int)j+10); //j를 int로 강제 변환 출력값 20

//byte<short<int<long<float<double

작은 타입에서 큰 타입은 자동 변환 되지만

큰 타입에서 작은타입은 강제변환 없으면 에러난다.

기본적으로 같은 타입형 변수만이 연산 가능하다.

다른 타입끼리 연산이 가능한데 이때는 작은 타입이 큰 타입으로 자동변환되어서 연산이 가능하다

하지만 큰타입에서 작은 타입으로 변환이 되지 않으므로 강제 변환을 통해서 연산을 해주어야 원하는 결과값을 얻는다.

Test007

정수값의 대표형 : int

실수값의 대표형 : double