

1. 자바시작

예제1)

```
class Hello
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

예제2)

```
class FirstApp
{
    public static void main(String[] args)
    {
        System.out.println("Hello World!");
    }
}
```

2. 자료형

예제3)

```
public class Data01 {
    public static void main(String[] args) {
        //(1) 정수 : 소수점이 없는 수
        System.out.println(1);
        //(2) 실수 : 소수점이 있는 수
        System.out.println(1.5);
        //(3) 문자 : 단일 따옴표로 묶어줌
        System.out.println('a');
        //(4) 논리값 : true, false
        System.out.println(true);
    }
}
```

예제4)

```
public class Data02 {

    public Data02() {
    }

    public static void main(String[] args) {
        // TODO code application logic here
        //(1) long 형 상수 : 숫자 끝에 L 혹은 l을 붙임
        System.out.println(1L);
        //(2) float 형 상수 : 숫자 끝에 F 혹은 f를 붙임
        System.out.println(1.5f);
        //(3) 문자열 : 이중 따옴표로 묶어줌
        System.out.println("abc");
    }
}
```

예제5)

```
public class Data03 {

    public Data03() {
    }

    public static void main(String[] args) {
        // TODO code application logic here
        //a=1; //error-선언된 변수가 없어서
        int a;//변수 선언하고
        a=1; //변수에 값을 저장
        System.out.println(a);
        //1=2; //error-상수는 값을 변경할 수 없다.
        a=2; //변수는 값을 변경할 수 있다.
        //마지막에 대입한 값만 유지됨
        System.out.println(a);
    }
}
```

예제6)

```
public class Data05 {
    public static void main(String[] args) {
        byte a=1;
        short b=128;
        int c=32768;

        b=a;                //암시적인 형 변환
        System.out.println(b);

        b=(short)c;          // 명시적인 형변환

        System.out.println(b); //오버플로우가 발생되어 엉뚱한 값 출력
    }
}
```

예제7)

```
public class Data06 {
    public static void main(String[] args) {
        double a=23.7;
        float b=23.7f;
        System.out.println(a);
        System.out.println(b);
    }
}
```

예제8)

```
public class Data07 {
    public static void main(String[] args) {
        char x;
        x='A';
        System.out.printf(" %c -> %d \n", x, (int)x);
        x='0';
        System.out.printf(" %c -> %d \n", x, (int)x);
        x=0;    //0은 NULL 문자를 의미
        System.out.printf(" %c -> %d \n", x, (int)x);
        x='a';
        System.out.printf(" %c -> %d \n", x, (int)x);
    }
}
```

예제9)

```
public class Data08 {
    public static void main(String[] args) {
        String y;
        y="AB";
        System.out.println(y);
        y="A";
        System.out.println(y);
    }
}
```

3. 연산자

예제10)

```
class Opr01 {
    public static void main(String[] args) {
        int a=10, b=4, c;
        c=a+b;
        System.out.println(a + " + " + b + " = " + c);
        c=a-b;
        System.out.println(a + " - " + b + " = " + c);
        c=a*b;
        System.out.println(a + " * " + b + " = " + c);
        c=a/b;
        System.out.println(a + " / " + b + " = " + c);
        c=a%b;
        System.out.println(a + " % " + b + " = " + c);
    }
}
```

예제11)

```
public class Opr01_02 {
    public static void main(String[] args) {
        String a="Apple";
        String b="Banana";
        String c=a+b;
        System.out.println(c);
        String str="결과값 : ";
        int n=10;
        System.out.println(str+n);
        System.out.println("결과값 : "+n);
    }
}
```

예제12)

```
public class Opr02 {
    public static void main(String[] args) {
        int a=10, b=4, c;
        boolean test;
        test=a>b;
        System.out.println(a + " > " + b + " = " + test);
        test=a<b;
        System.out.println(a + " < " + b + " = " + test);

        //두 산술연산자 결과에 대한 대소 관계를 물어볼 수도 있다.
        System.out.println(a+ b > a-b);
    }
}
```

예제13)

```
public class Opr03 {
    public static void main(String[] args) {
        int a=5, b=10;
        int max; //최대값을 저장할 변수 선언
        max = a>b ? a : b;
        System.out.println(" max = "+ max);
    }
}
```

예제14)

```
class Opr04 //대입 연산자와 증감 연산자
{
    public static void main(String[] args)
    {
        int a=29;
        String s="몰라";
        s = (a>=10 && a <=19) ? "10대" : "10대 아님";
        System.out.println(a + " => " + s);
    }
}
```

예제15)

```
class Opr06
{
    public static void main(String[] args)
    {
        char ch='b'; //문자 변수 선언
        String s;     //대문자인지 판단한 결과를 저장할 문자열 변수 선언

        s = (ch>='A' && ch<='Z') ? "대문자임" : "대문자가 아님";
        System.out.println(ch + "=>" + s);
    }
}
```

예제16)

```
class Opr07
{
    public static void main(String[] args)
    {
        int a=10, b=10;
        ++a;          //선행처리
        System.out.println(a);
        b++;          //후행처리
        System.out.println(b);
    }
}
```

예제17)

```
class Opr08 //대입 연산자와 증감 연산자
{
    public static void main(String[] args)
    {
        int a=10, b=10;
        System.out.println(++a);
        System.out.println(a);
        System.out.println(b++);
        System.out.println(b);

        a=b=10;
        int c;
        c=++a;
        System.out.println(c + " => " + a);
        c=b++;
        System.out.println(c + " => " + b);
    }
}
```

예제18)

```
class Opr09 //비트 단위 논리 연산자
{
    public static void main(String[] args)
    {
        int a=12; // 8+4 = 23 + 22
        int b=20; //16+4= 24 + 22
        int c;
        c = a & b; //비트 단위 값이 둘 다 1일 때만 1
        System.out.println(a + " & " + b + " -> " + c);

        c = a | b; //비트 단위 값이 둘 다 0일때만 0
        System.out.println(a + " | " + b + " -> " + c);

        c = a ^ b; //비트 단위 값이 두 값이 다르면 1 같으면 0
        System.out.println(a + " ^ " + b + " -> " + c);

        c = ~a; //비트 단위 값이 1이면 0으로 0이면 1로
        System.out.println("~" + a + " -> " + c);
    }
}
```

예제19)

```
class Opr10 {
    public static void main(String[] args) {
        byte x = 15;

        System.out.println("x << 2 : " + (x << 2) );
        System.out.println("x >> 2 : " + (x >> 2) );
    }
}
```

4. 선택문

예제20)

```
class If01 {  
    public static void main(String[] args) {  
        int num;  
        num = -5;  
  
        if(num < 0)  
            num = -num;  
  
        System.out.println(" absolute num = "+ num);  
        num = 5;  
  
        if(num < 0)  
            num = -num;  
  
        System.out.println(" absolute num = "+ num);  
    }  
}
```

예제21)

```
class If02 {  
    public static void main(String[] args) {  
        int num=Integer.parseInt(args[0]); //num=40;  
  
        if(num%2==1) //정수형 수치 데이터를 2로 나누어서 나머지가 1이면  
            System.out.println(num + "는 홀수입니다.");  
        else //정수형 수치 데이터를 2로 나누어서 나머지가 0이면  
            System.out.println(num + "는 짝수입니다.");  
    }  
}
```


예제22)

```
class If03 {
    public static void main(String[] args) {
        int a=Integer.parseInt(args[0]);    // a=5
        int b=Integer.parseInt(args[1]);    // b=10
        int max, min;    //최대값, 최소값을 저장할 변수 선언

        if(a > b){        //조건에 만족할 경우에 수행할 문장이 2개 이상이면
            max=a;        //반드시 {}로 문장들을 묶어주어야 한다.
            min=b;
        }
        else{            //조건에 만족하지 않은 경우에도 수행할 문장이 2개 이상이면
            max=b;        //반드시 {}로 문장들을 묶어주어야 한다.
            min=a;
        }
        System.out.println("최대 값은 " + max + "입니다.");
        System.out.println("최소 값은 " + min + "입니다.");
    }
}
```

예제23)

```
class If04 {
    public static void main(String[] args) {
        int a=Integer.parseInt(args[0]);//a=5;

        if(a>0)        //a가 0보다 크면
            System.out.println(a+ "는 양수입니다.");
        else if(a<0)    //a가 0보다 작으면
            System.out.println(a+ "는 음수입니다.");
        else            //a가 0보다 크지도 작지도 않으면
            System.out.println("0 입니다.");
    }
}
```

예제24)

```
class If05 {
    public static void main(String[] args){
        int score=Integer.parseInt(args[0]); // score=85;

        System.out.print (score + "는 ");

        if(score >= 90 && score <= 100)
            System.out.println("A학점");
        else if(score >= 80 && score <= 89)
            System.out.println("B학점");
        else if(score >= 70 && score <= 79)
            System.out.println("C학점");
        else if(score >= 60 && score <= 69)
            System.out.println("D학점");
        else
            System.out.println("F학점");
    }
}
```

예제25)

```
class Switch01{
    public static void main(String[] args){
        int a = Integer.parseInt(args[0]);

        System.out.println(" => " + a);

        switch(a){
            case 9: System.out.println( " A ");
            case 8: System.out.println( " B ");
            case 7: System.out.println( " C ");
            case 6: System.out.println( " D ");
            default : System.out.println( " F ");
        }
    }
}
```

예제26)

```
class Switch02 {
    public static void main(String[] args) {
        int a=Integer.parseInt(args[0]);

        System.out.println(" => " + a);

        switch(a){
            case 9: System.out.println( " A "); break;
            case 8: System.out.println( " B "); break;
            case 7: System.out.println( " C "); break;
            case 6: System.out.println( " D "); break;
            default : System.out.println( " F ");
        }
    }
}
```

예제27)

```
class Switch03 {
    public static void main(String[] args) {
        int a=Integer.parseInt(args[0]);

        System.out.println(" => " + args[0]);
        a=a/10;    // 점수를 입력받아 그 점수를 10으로 나누어 몫을 구한다.

        switch(a){ // 구해진 몫이 무엇이냐에 따라 학점을 출력한다.
            case 10:
            case 9: System.out.println( " A "); break;
            case 8: System.out.println( " B "); break;
            case 7: System.out.println( " C "); break;
            case 6: System.out.println( " D "); break;
            default : System.out.println( " F ");
        }
    }
}
```

예제28)

```
class Switch04 {
    public static void main(String[] args) {
        String str = args[0];
        char ch;
        ch=str.charAt(0);    //='b';

        switch(ch) {
            case 'A' :
            case 'a' : System.out.println("America"); break;
            case 'B' :
            case 'b' : System.out.println("Britain"); break;
            case 'C' :
            case 'c' : System.out.println("Canada"); break;
            case 'J' :
            case 'j' : System.out.println("Japan"); break;
            case 'K' :
            case 'k' : System.out.println("Korea"); break;
        }
    }
}
```

예제29)

```
class Test01 {
    public static void main(String [] args){
        System.out.println("입력받은 문자열의 개수="+ args.length);

        for(int i=0 ; i < args.length ; i++)
            System.out.println("args[ " + i + " ]= " + args[i]);
    }
}
```

5. 반복문

예제30)

```
class For01
{
    public static void main(String[] args)
    {
        int i;
        for(i=1; i<=10; i++)
            System.out.println("Hello World!");
    }
}
```

예제31)

```
class For02{
    public static void main(String[] args) {
        int i;

        for(i=1; i<=4; i++)
            System.out.println(i );    //제어변수 i값을 출력

        System.out.println("--->> " + i );
    }
}
```

예제32)

```
class For03{
    public static void main(String[] args) {
        int i;

        // i의 초기값을 1로 하여 1씩 증가하면서 10일 때까지 반복
        for(i=1; i<=10; i++)
            System.out.print(" " + i );
        System.out.println("\n ----- >> " );

        // i의 초기값을 1로 하여 2씩 증가하면서 10일 때까지 반복
        for(i=1; i<=10; i+=2)
            System.out.print(" " + i );
        System.out.println("\n ----- >> " );

        // i의 초기값을 2로 하여 2씩 증가하면서 10일 때까지 반복
        for(i=2; i<=10; i+=2)
            System.out.print(" " + i );
        System.out.println("\n ----- >> " );

        // i의 초기값을 10으로 하여 1씩 감소하면서 1보다 크거나 같을 때까지 반복
        for(i=10; i>=1; i--)
            System.out.print(" " + i );
        System.out.println("\n ----- >> " );
    }
}
```

예제33)

```
class For04 {
    public static void main(String[] args) {
        int i;    //제어변수 선언
        int a=2; //출력할 단을 저장하는 변수 선언, 2단 출력

        System.out.println("<<-----" + a + "단----->> " );
        for(i=1; i<=9; i++)
            System.out.println(a + " * " + i + " = " + (a * i) );
    }
}
```

예제34)

```
class For04_1 {
    public static void main(String[] args) {
        int i, n=0;
        int a=Integer.parseInt(args[0]);//출력하고자하는 단을 명령행에서 5를 입력 받았다면

        System.out.println("<<-----" + a + "단----->> " );

        for(i=1; i<= 9; i++){
            n=a*i;
            System.out.println(a + " * " + i + " = " + n );
        }
    }
}
```

예제35)

```
class For05{
    public static void main(String[] args) {
        int i;          //제어변수
        int repeat; //반복 횟수를 결정할 변수

        String outStr=args[0];
        repeat = Integer.parseInt(args[1]);

        for(i=1; i<=repeat; i++)
            System.out.print(" " + outStr);
    }
}
```

예제36)

```
class For06{
    public static void main(String[] args) {

        int i;                //제어변수 선언
        int total = 0;        //합을 누적할 변수 total을 선언하고 0으로 초기화

        for(i=1; i<=5; i++) //제어변수 i이 1부터 5까지 1씩 증가하도록 함
            total +=i;       // total = total + i;

        System.out.println("1 ~ " + (i-1) + " = " + total);

    }
}
```

예제37)

```
class For07{
    public static void main(String[] args) {
        int i;           //제어변수 선언
        int n;
        int fac = 1;     //팩토리알을 누적할 변수 선언 시 초기값을 1로 줌

        n = Integer.parseInt(args[0]); //키보드에서 5를 입력받았다면

        for(i=1; i<=n; i++)          //1부터 5사이의 자연수를 구함
            fac*=i; //fac=fac*i;      //자연수를 곱해서 누적하여 팩토리알을 구함

        System.out.println( n + "! ----- >> " + fac );
    }
}
```

예제38)

```
class While01
{
    public static void main(String[] args)
    {
        int i;

        for(i=1; i<=5; i++)
            System.out.print(" " + i);

        System.out.println("\n-----");

        i=1;                                //초기식

        while(i<=5){                         //조건식
            System.out.print(" " + i);        //반복 처리할 문장
            i++;                             //증감식
        }
    }
}
```


예 제39)

```
class While02{
    public static void main(String[] args)
    {
        int i=1;
        while(i++ <=4)
            System.out.print( i + ", " );
        System.out.println("\n----- >>" );

        i=1;
        while(++i <=4)
            System.out.print( i + ", " );
        System.out.println("\n----- >>" );

        i=0;
        while(i++ <=4)
            System.out.print( i + ", " );
        System.out.println("\n----- >>" );
    }
}
```

예 제40)

```
class While03{
    public static void main(String[] args) {
        int i=0;
        while(i++ <=4)
            System.out.print( i + ", " );
        System.out.println("\n----- >>" );

        i=1;
        do
            System.out.print( i + ", " );
        while(i++ <=4);           //do while 문은 반드시 세미콜론으로 끝난다.
        System.out.println("\n----- >>" );

        i=0;
        while(i++ <=4) ;           //NULL 문으로 인식함
        System.out.print( i + ", " );
        System.out.println("\n----- >>" );

        for(i=1; i<=4; i++) ;       //NULL 문으로 인식함
        System.out.print( i + ", " );
        System.out.println("\n-----");
    }
}
```

예제41)

```
class While04 {
    public static void main(String[] args) {
        int n;           //제어변수 선언
        int tot=0;        //합을 누적할 변수 선언

        n=0;              //제어변수 n를 0으로 초기화
        while(n<=8){      //제어변수 n이 8보다 작거나 같을 때까지 반복 수행
            n+=2;          //제어변수 n을 2씩 증가한 후
            tot += n;       //제어변수 n을 합을 누적할 변수에 더한다.
        }
        System.out.println("tot = "+ tot); //반복문에서 벗어나면 합을 출력한다.
    }
}
```

6. 제어문 활용

예제42)

```
class E01 {
    public static void main(String[] args) {
        int n;           //제어변수 선언
        int odd_tot, even_tot; //홀수의 합과 짝수의 합을 누적할 변수

        for(odd_tot=0 , n=1; n<=10; n+=2) //홀수를 구함
            odd_tot += n;                 //구해진 홀수를 누적

        for(even_tot=0 , n=2; n<=10; n+=2) //짝수를 구함
            even_tot += n;                 //구해진 짝수를 누적

        System.out.println(" odd_tot(1+ 3+ 5+ 7+ 9) = " + odd_tot); //홀수의 합을 출력
        System.out.println(" even_tot(2+ 4+ 6+ 8+ 10) = " + even_tot); //짝수의 합을 출력
    }
}
```

예제43)

```
class E01_01 {
    public static void main(String[] args) {
        int n;                //제어변수 선언
        int odd_tot, even_tot; //홀수의 합과 짝수의 합을 누적할 변수

        for(odd_tot=0, even_tot=0 , n=1; n<=10; n++) //제어변수 n은 1부터 10사이의 자연수
            if(n%2==1)                //n을 2로 나누어서 나머지가 1이면 홀수이므로
                odd_tot += n;          //홀수의 합을 누적하는 변수에 더하고
            else                        //n을 2로 나누어서 나머지가 1이 아니고 0이면 짝수이므로
                even_tot += n;         //짝수의 합을 누적하는 변수에 더한다.

        System.out.println(" odd_tot(1+ 3+ 5+ 7+ 9) = " + odd_tot);    //홀수의 합을 출력
        System.out.println(" even_tot(2+ 4+ 6+ 8+ 10) = " + even_tot); //짝수의 합을 출력
    }
}
```

예제44)

```
public class E04 {
    public static void main(String[] args) {
        int a;                //바깥쪽 for문의 제어변수 선언
        int b;                //안쪽 for문의 제어변수 선언

        for(a=1; a<=5; a++){   //5줄 반복한다.
            for(b=1; b<=5; b++){ //한 줄에 스타(*)를 5번 출력하기 위한 반복문
                System.out.print("* "); //안쪽 for문에 의해 반복되는 문장
            } //안쪽 for문의 끝
            System.out.println();       //줄 바꾸기 위한 문장
        } //바깥쪽 for문의 끝

    } //main 함수의 끝
}
```

예제45)

```
public class E05 {
    public static void main(String[] args) {
        int line;           //라인수를 결정하는 변수
        int spc;             //공백을 몇 번 출력할지 결정하는 변수
        int n;               //숫자를 몇 번 출력할지 결정하는 변수
        int number=1;        //출력할 숫자를 저장하는 변수
        int size=3;

        for(line=1; line<=size ; line++){           //3줄로 출력
            for(spc=size-line; spc>=1 ;spc--)        //공백 출력할 횟수를 결정
                System.out.print(" ");              //공백 출력
            for(n=1; n<= line*2-1; n++)               //숫자 출력할 횟수를 결정
                System.out.print(number++);          //숫자 출력
            System.out.println("");                  //줄바꿈
        }

        }//main 함수의 끝
    }
```

예제46)

```
class E08{
    public static void main(String[] args) {
        int i;
        int a;

        System.out.println("시침 ----->> 분침");
        System.out.println("a(바깥 쪽 제어변수)-->> i(안 쪽 제어변수)");

        for(a=1; a<5; a++)
            for(i=1; i<=5; i++)
                System.out.println(a + "----- >> " + i );

        }
    }
```

예제47)

```
class F01 {
    public static void main(String[] args) {
        int n;

        for(n=1; n<=10; n++)          //1부터 10까지 자연수를 구함
            System.out.print(" " + n);

        System.out.println("");

        for(n=1; n<=10; n++){          //1부터 10까지 자연수를 구함
            if(n%3==0)                  //제어변수 n이 3으로 나누어 떨어지면
                break;                  //반복문을 벗어남

            System.out.print(" " + n);
        }
        System.out.println("");

    } //main 함수의 끝
}
```

예제48)

```
class F02 {
    public static void main(String[] args) {
        int n;

        for(n=1; n<=10; n++){          //1부터 10까지 자연수를 구함
            if(n%3==0)                  //제어변수 n이 3으로 나누어떨어지면
                continue;              //증감식으로 제어가 옮겨져서 반복은 계속됨

            System.out.print(" " + n);
        }
        System.out.println("");

    } //main 함수의 끝
}
```

예제49)

```
class F03 {
    public static void main(String[] args) {
        int n=0;

        while( true ) {
            System.out.print("Fall Wonderland ");

            if(++ n >=10)
                break;
        }
        System.out.println("The End ");
    }
}
```

예제50)

```
class F04 {
    public static void main(String[] args) {
        int n=0;
        for( ; ; ) {
            System.out.print("Fall Wonderland ");
            if(++ n >=10)
                break;
        }
        System.out.println("The End ");
    }
}
```

예제51)

```
class F10{
    public static void main(String[] args)
    {
        int i;
        int a;

        for(a=1; a<10; a++){
            for(i=1; i<=10; i++){
                if(i%3==0)
                    break;

                System.out.print("  i->" + i );
            }
            System.out.println("\n a -> " + a);
        }
        System.out.println("-----\n");

        exit_for:
        for(a=1; a<10; a++){
            for(i=1; i<=10; i++){
                if(i%3==0)
                    break exit_for;

                System.out.print("  i->" + i );
            }
            System.out.println("\n a -> " + a);
        }
        System.out.println("\n-----");
    }
}
```

7. 배열

예제52)

```
public class Arr01 {
    public static void main(String[] args) {
        int []score = {95, 70, 80, 75, 100};
        int total=0;
        double ave;

        //반복문으로 배열을 일괄 처리함
        for(int i=0; i<5; i++)
            total += score[ i ];      //총합을 구함

        ave = (double) total / 5.0;  //평균을 구함

        System.out.println(" Total = " + total);    //총합 출력
        System.out.println(" Ave   = " + ave);      //평균 출력
    }
}
```

예제53)

```
public class Arr02 {
    public static void main(String[] args) {
        double []data = {9.5, 7.0, 13.6, 7.5, 10.0};
        double max;    //반복문을 수행하기 전에 첫 번째(첨자가 0인) 데이터를 최대값으로 설정
        max = data[0];

        for (int i = 1; i < 5 ; i++)
            if(data[i] > max)    //배열의 원소가 최대값보다 크면
                max = data[i];  //새로운 최대값으로 설정

        System.out.println("  max = " + max);  //최대값 출력
    }
}
```


예제54)

```
public class Arr03 {
    public static void main(String[] args) {
        //정수값을 담을 수 있는 5행 3열짜리 기억 공간이 생성
        int [][]score=new int [5][3];
        int row, col; //반복문에서 사용할 제어변수 선언

        //행과 열의 위치를 첨자로 지정하여 값 대입
        score[0][0]=10; score[0][1]=90; score[0][2]=70;
        score[1][0]=60; score[1][1]=80; score[1][2]=65;
        score[2][0]=55; score[2][1]=60; score[2][2]=85;
        score[3][0]=90; score[3][1]=75; score[3][2]=95;
        score[4][0]=60; score[4][1]=30; score[4][2]=80;

        ///반복문으로 일괄처리
        for(row = 0; row < 5 ; row++){
            for(col = 0; col < 3 ; col++){
                System.out.print(" " + score[row][col]);

                System.out.println(""); //행단위로 줄 바꿈
            }
        }
    }
}
```

예제55)

```
public class Arr04 {
    public static void main(String[] args) {
        //학생들의 과목별 점수를 저장한 2차원 배열 선언
        int [][]score = { { 85, 60, 70},    //0 행
                           { 90, 95, 80},    //1 행
                           { 75, 80, 100},    //2 행
                           { 80, 70, 95},    //3 행
                           {100, 65, 80}     //4 행
                        };

        int [] subject = new int[3]; //각 과목별 총점을 저장할 변수 선언
        int [] student = new int[5]; //각 학생별 총점을 저장할 변수 선언

        int r, c;

        System.out.println("각 과목별 총점구하기 ");
        for(c = 0; c < 3 ; c++){           //열은 각 과목을 구분하고 해당 과목에 대해서
            for(r = 0; r < 5 ; r++){         //행을 증가해 가면서
                subject[c] += score[r][c];  //각 과목별 총점을 구한다.
            }
            System.out.println(subject[c]); //각 과목별 총점 출력하기
        }

        System.out.println("학생별 총점구하기");
        for(r = 0; r < 5 ; r++){           //행은 각 학생을 구분하고 해당 학생에 대해서
            for(c = 0; c < 3 ; c++){         //열을 증가해 가면서
                student[r] += score[r][c];  //국어, 영어, 수학 점수를 합하여 학생별 총점을 구한다.
            }
            System.out.println(student[r]); //학생별 총점 출력하기
        }
    }
}
```

예제56)

```
public class G01 {
    public static void main(String[] args) {
        int []score = new int [5]; //5명의 점수를 저장하기 위한 배열 선언

        //배열의 원소에 접근하여 점수를 저장
        score[0]=95;
        score[1]=70;
        score[2]=80;
        score[3]=75;
        score[4]=100;

        //반복문으로 배열을 일괄 처리함
        for(int i=0; i<5; i++)
            System.out.println( (i+1) + " th score[ " + i + " ] = " + score[i]);
    }
}
```

예제57)

```
public class G02 {
    public static void main(String[] args) {
        //각 달의 날짜수를 초기값으로 설정
        int [] month = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,31};

        for(int i=0; i<12; i++)//반복문으로 배열을 일괄 처리함
            System.out.println( (i+1) + " month => " + month[i]);
    }
}
```

8. 메소드

예제58)

```
public class MethodEx01 {
    static void hello_func(){
        System.out.println("Hello World!");
    }
    public static void main(String[] args) {
        //사용자 정의 함수 호출
        hello_func();
    }
}
```

예제59)

```
public class MethodEx02 {

    //n은 메소드를 호출할 때 넘어온 값을 담아두기 위한 기억
    static void sum(int n){           //이를 형식매개변수라 함
        int i;                       //제어변수 선언
        int tot=0;                   //합을 누적할 변수 선언

        for(i=1; i<=n; i++)          //1부터 메소드로 전달된 값까지의 자연수를 구함
            tot += i;                //합을 누적

        System.out.println("1 ~ "+ n + " = " + tot); //구해진 합을 출력
    }

    public static void main(String[] args) {
        sum(5);                      //메소드를 호출할 때 실매개변수로 5를 넘겨줌
        sum(10);                     //메소드를 호출할 때 실매개변수로 10을 넘겨줌
    }
}
```

예제60)

```
public class MethodEx03 {

    //정수값을 하나 전달받음
    static int abs(int data){
        if(data<0)                   //전달받은 정수값이 음수라면
            data = -data;            //-연산자를 이용하여 양수로 변경

        return data;                 //결과값을 되돌립니다.
    }

    public static void main(String[] args) {
        int num=Integer.parseInt(args[0]);

        System.out.println(" absolute data => "+ abs(num));
    }
}
```

9. 클래스와 객체

예제61)

```
public class AnimalTest01 {
    public static void main(String[] args) {
        Animal a1;                //레퍼런스 변수 선언

        a1=new Animal();           //객체 생성
        a1.name="원숭이";           //생성된 객체의 멤버에 접근해서 값 대입
        a1.age=26;

        System.out.print(a1.name); //객체의 멤버에 저장된 값 출력
        System.out.print(", "+ a1.age);

    }
}

//이름과 나이를 속성으로 갖는 클래스 설계
class Animal {
    /** 속성 선언 */
    String name; //이름 속성
    int age;     //나이 속성
}
```

예제62)

```
public class AnimalTest02 {
    public static void main(String[] args) {
        Animal a1;

        a1=new Animal();
        a1.name="원숭이";
        a1.age=26;                //private 멤버이므로 접근 불가능

        System.out.print(a1.name);
        System.out.print(", "+ a1.age); //private 멤버이므로 접근 불가능
    }
}

class Animal {
    String name;
    private int age;
}
```

예제63)

```
class Animal {
    String name;
    private int age;

    public void setAge(int new_age){
        age=new_age;
    }
    public int getAge(){
        return age;
    }
}

public class AnimalTest03 {
    public static void main(String[] args) {
        Animal a1;
        a1=new Animal();
        a1.name="원숭이";
        //a.age=26;                      //public이어야만 허용된다.
        a.setAge(26);
        System.out.println(a.name);
        //System.out.println(", "+ a.age);    //public이어야만 허용된다.
        System.out.println(", " + a.getAge( ) );
    }
}
```

10. 메소드 활용

예제64)

```
public class MethodTest01{
    public static void main(String[] args) {
        //(1) 논리값 : true, false
        System.out.println(true);
        //(2) 문자 : 단일 따옴표로 묶어줌
        System.out.println('A');
        //(3) 정수 : 소수점이 없는 수
        System.out.println(128);
        //(4) 실수 : 소수점이 있는 수
        System.out.println(3.5);
        //(5) 문자열 : 이중 따옴표로 묶어줌
        System.out.println("Hello");
    }
}
```

예제65)

```
public class MethodTest02{
    //int형 데이터에 대해서 절대값을 구하는 메소드 정의
    int abs(int num){
        if(num<0)
            num=-num;
        return num;
    }

    //long형 데이터에 대해서 절대값을 구하는 메소드 정의
    long abs(long num){
        if(num<0)
            num=-num;
        return num;
    }

    //double 데이터에 대해서 절대값을 구하는 메소드 정의
    double abs(double num){
        if(num<0)
            num=-num;
        return num;
    }

    public static void main(String[] args) {
        MethodTest02 mt=new MethodTest02();

        //전달인자가 int형이므로 03:의 int형 데이터에 대해서 절대값을 구하는 메소드 호출
        int var01=-10, var02;
        var02=mt.abs(var01);
        System.out.println(var01 + "의 절대값은-> " + var02);

        //전달인자가 long형이므로 09:의 long형 데이터에 대해서 절대값을 구하는 메소드 호출
        long var03=-20L, var04;
        var04=mt.abs(var03);
        System.out.println(var03 + "의 절대값은-> " + var04);

        //전달인자가double형이므로 15:의 double형 데이터에 대해서 절대값을 구하는 메소드 호출
        double var05=-3.4, var06;
        var06=mt.abs(var05);
        System.out.println(var05 + "의 절대값은-> " + var06);
    }
}
```

예제66)

```
public class MethodTest03{
    //정수형 데이터 3개를 형식매개변수로 갖는 prn 메소드 정의
    void prn(int a, int b, int c) {
        System.out.println(a + "Wt" + b + "Wt" + c);
    }
    //정수형 데이터 2개를 형식매개변수로 갖는 prn 메소드 정의
    void prn(int a, int b) {
        System.out.println(a + "Wt" + b);
    }
    //정수형 데이터 1개를 형식매개변수로 갖는 prn 메소드 정의
    void prn(int a){
        System.out.println(a);
    }
    public static void main(String[] args){
        MethodTest03 mt=new MethodTest03();
        mt.prn(10, 20, 30); //정수형 데이터 3개를 실매개변수로 지정
        mt.prn(40, 50);    //정수형 데이터 2개를 실매개변수로 지정
        mt.prn(60);        //정수형 데이터 1개를 실매개변수로 지정
    }
}
```

예제67)

```
public class MethodTest04{
    void prn(int ... num){    //int형 데이터를 출력하는 메소드의 정의
        for(int i=0; i<num.length; i++) //전달인자의 개수만큼 반복하면서
            System.out.print(num[i]+ "Wt"); //배열 형태로 출력한다.

        System.out.println();
    }

    public static void main(String[] args)
    {
        MethodTest04 mt=new MethodTest04();
        mt.prn(10, 20, 30); //개수에 상관없이 메소드를 호출할 수 있다.
        mt.prn(40, 50);
        mt.prn(60);
    }
}
```


예제68)

```
class MyDate{    //클래스의 초기값을 지정함
    int year=2006;
    int month=4;
    int day=1;
}

public class MethodTest05 {
    public static void main(String[] args) {

        MyDate d;    //1. 레퍼런스 변수
        System.out.println(d.year+ "/" + d.month+ "/" + d.day);

        new MyDate(); //2. 객체 생성되었지만 사용되지 못함

        d=new MyDate();
        System.out.println(d.year+ "/" + d.month+ "/" + d.day);
    }
}
```

예제69)

```
class MyDate{
    int year=2006;
    int month=4;
    int day=1;
}

public class MethodTest06{
    public static void main(String[] args) {
        //기본 자료형 중에서 int형으로 선언한 두 개의 변수
        int x=7;
        int y=x;                //7을 저장하고 있는 변수 x의 값을 변수 y에 복사

        //두 개는 MyDate 형으로 선언된 레퍼런스 변수
        MyDate d=new MyDate();    //객체 생성
        MyDate t=d;

        System.out.println( "x->" + x+ " y->" + y);
        System.out.println(d.year+ "/" + d.month+ "/" + d.day);    //2006/4/1
        System.out.println(t.year+ "/" + t.month+ "/" + t.day);    //2006/4/1

        //변수 x와 y는 독립된 두 개의 변수이므로
        y=10;//변수 y의 값을 변경시켜도 x의 값에 영향을 주지 못함
        System.out.println( "x->" + x+ " y->" + y);

        //레퍼런스 변수 t로 접근해서 MyDate 객체의 값을 변경하면
        t.year=2007;    t.month=7;    t.day=19;

        //레퍼런스 변수 d로 접근했을 때에도 변경되어진 값이 출력
        System.out.println(d.year+ "/" + d.month+ "/" + d.day);    //2007/7/19
        System.out.println(t.year+ "/" + t.month+ "/" + t.day);    //2007/7/19
    }
}
```

예제70)

```
class MyDate{
    int year=2006;
    int month=4;
    int day=1;
}

public class MethodTest07{
    public static void main(String[] args) {
        MyDate d=new MyDate(); //객체 생성
        MyDate t=d;           //t가 이미 선언된 d와 동일한 객체를 참조함

        System.out.println(d.year+ "/" + d.month+ "/" + d.day); //2006/4/1
        System.out.println(t.year+ "/" + t.month+ "/" + t.day);  //2006/4/1

        t=new MyDate();      //t가 새로 생성된 객체를 가리키므로 d와는 별개로 동작함
        t.year=2007;   t.month=7;   t.day=19;

        System.out.println(d.year+ "/" + d.month+ "/" + d.day); //2006/4/1
        System.out.println(t.year+ "/" + t.month+ "/" + t.day);  //2007/7/19
    }
}
```

예제71)

```
class ValueMethod{
    void changeInt(int y){
        y=10;
    }
}

public class MethodTest08 {
    public static void main(String[] args) {
        ValueMethod vm=new ValueMethod();
        int x=7;

        System.out.println( " 함수 호출 전 x->" + x);
        vm.changeInt(x);
        System.out.println( " 함수 호출 후 x->" + x);
    }
}
```

예제72)

```
class MyDate{
    int year=2006;
    int month=4;
    int day=1;
}

class RefMethod{
    void changeDate(MyDate t){
    }
}

public class MethodTest09 {
    public static void main(String[] args) {
        RefMethod rm=new RefMethod();
        MyDate d=new MyDate();

        System.out.println(" 함수 호출 전 d->" + d.year+ "/" + d.month+ "/" + d.day);
        rm.changeDate(d);
        System.out.println(" 함수 호출 후 d->" + d.year+ "/" + d.month+ "/" + d.day);
    }
}
```

예제73)

```
class MethodTestA{
    void prn(int a){
        System.out.println(a);
    }

    int prn(int a){
        return a;
    }

    public static void main(String[] args){
        MethodTestA mt=new MethodTestA();

        mt.prn(10);          //정수형 데이터 1개를 실매개변수로 지정
        int k;
        k=mt.prn(10);
    }
}
```

예제74)

```
class MyDate{
    int year=2006;
    int month=4;
}

class MethodTestB{
    public static void main(String[] args){
        MyDate d;
        System.out.println(d.year+ "/" + d.month+ "/" + d.day);
    }
}
```

예제75)

```
class MyDate{
    int year=2006;
    int month=4;
    int day=1;
}

class MethodTestC{
    public static void main(String[] args){
        MyDate d=null;
        System.out.println(d.year+ "/" + d.month+ "/" + d.day);
        System.out.println("정상 종료");
    }
}
```

예제76)

```
class MyDate{
    int year=2006;
    int month=4;
    int day=1;
}

class MethodTestD{
    public static void main(String[] args){
        MyDate d=null;
        try{
            System.out.println(d.year+ "/" + d.month+ "/" + d.day);
        }catch(Exception e){
            System.out.println("예외 발생");
        }
        System.out.println("정상 종료");
    }
}
```

11. 생성자

예제77)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest01 {
    public static void main(String[] args) {
        MyDate d=new MyDate();    //디폴트 생성자 호출
        d.print();
    }
}
```

예제78)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    public MyDate(){
        System.out.println("[생성자] : 객체가 생성될 때 자동 호출됩니다.");
    }

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest02 {
    public static void main(String[] args) {
        MyDate d = new MyDate();
        d.print();
    }
}
```

예제79)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    public MyDate(){
        year=2006;
        month=4;
        day=1;
    }
    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest03 {
    public static void main(String[] args) {
        MyDate d=new MyDate();
        d.print();
    }
}
```

예제80)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    //생성자는 속성(멤버변수)들의 초기화 작업을 목적으로 한다.

    //[1] 전달인자 없는 생성자 정의
    public MyDate(){
        year=2006;    month=4;    day=1;
    }

    //[2] 전달인자 있는 생성자 정의
    public MyDate(int new_year, int new_month, int new_day){
        year=new_year;
        month=new_month;
        day=new_day;
    }

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest04 {
    public static void main(String[] args) {
        MyDate d=new MyDate();
        d.print();

        MyDate d2=new MyDate(2007, 7, 19);
        d2.print();
    }
}
```


예제81)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    public MyDate(int new_year, int new_month, int new_day){
        year=new_year;
        month=new_month;
        day=new_day;
    }

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest05 {
    public static void main(String[] args) {
        MyDate d=new MyDate();
        d.print();

        MyDate d2=new MyDate(2007, 7, 19);
        d2.print();
    }
}
```

예제82)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    //생성자 정의하기
    public MyDate(){
    }

    public MyDate(int new_year, int new_month, int new_day){
        year=new_year;    month=new_month;    day=new_day;
    }

    //전달인자가 객체 속성의 이름과 동일한 메서드
    public void SetYear(int year){
        //this.year=year;
        year=year;
    }

    public void SetMonth(int new_month){
        month=new_month;
    }

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest06 {
    public static void main(String[] args) {
        MyDate d=new MyDate(2007, 7, 19);

        d.print();
        d.SetYear(2008);    //변경되지 않음

        d.print();        //-----?
        d.SetMonth(8);    //변경됨
        d.print();        //-----?
    }
}
```

```

class MyDate{
    private int year;
    private int month;
    private int day;

    public MyDate(){
    }

    //생성자 역시 매개변수의 이름을 속성과 동일하게 줄 수 있다.
    public MyDate(int year, int month, int day){
        //멤버변수로 속성 값을 초기화하려면 대입연산자 왼쪽에 this를 붙여야 한다.
        this.year=year;
        this.month=month;
        this.day=day;
    }

    public void SetYear(int year){ //대입연산자 왼쪽에 this를 붙였기에
        this.year=year;           //속성 값이 변경됨
    }

    public void SetMonth(int month){//대입연산자 왼쪽에 this를 붙였기에
        this.month=month;         //속성 값이 변경됨
    }

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest07 {
    public static void main(String[] args) {
        MyDate d=new MyDate(2007, 7, 19);

        d.print();
        d.SetYear(2008); //2008년으로 변경
        d.SetMonth(8);   //8월로 변경
        d2.print();
    }
}

```

예제84)

```
class MyDate{
    private int year;
    private int month;
    private int day;

    public MyDate(){
        this(2006, 1, 1);          //14:에 정의된 생성자 호출
    }

    public MyDate(int new_year){
        this(new_year, 1, 1);      //14:에 정의된 생성자 호출
    }

    public MyDate(int new_year, int new_month){
        this(new_year, new_month, 1); //14:에 정의된 생성자 호출
    }

    public MyDate(int new_year, int new_month, int new_day){
        year=new_year;
        month=new_month;
        day=new_day;
    }

    public void print(){
        System.out.println(year+ "/" + month+ "/" + day);
    }
}

public class ConstructorTest10 {
    public static void main(String[] args) {
        MyDate d=new MyDate(2007, 7, 19); //14:에 정의된 생성자 호출
        d.print();

        MyDate d2=new MyDate(2007, 7);    //11:에 정의된 생성자 호출
        d2.print();

        MyDate d3=new MyDate(2007);       //8:에 정의된 생성자 호출
        d3.print();

        MyDate d4=new MyDate();           //5:에 정의된 생성자 호출
        d4.print();
    }
}
```

12. static 활용

예제85)

```
class StaticTest{
    static int a=10;
    int b=20;
}

class StaticTest01 {
    public static void main(String[] args){

        System.out.println("StaticTest.a->" + StaticTest.a);
        StaticTest s1 = new StaticTest();
        StaticTest s2 = new StaticTest();

        System.out.println("s1.a->" + s1.a + " \t s2.a->" + s2.a);
        System.out.println("s1.b->" + s1.b + " \t s2.b->" + s2.b);

        s1.a=100;
        System.out.print("s1.a->" + s1.a );
        System.out.println("\t s2.a->" + s2.a);

        s1.b=200;
        System.out.print("s1.b->" + s1.b);
        System.out.println("\t s2.b->" + s2.b);
    }
}
```

예제86)

```
class StaticTest{
    private static int a=10;
    private int b=20;

    public static void setA(int new_a){
        a = new_a;
    }
    public static int getA(){
        return a;
    }
}
public class StaticTest02 {
    public static void main(String[] args) {
        //System.out.println(StaticTest.a);//a가 private으로 선언되어서 컴파일 에러 발생
        System.out.println(StaticTest.getA());

        StaticTest s1=new StaticTest();
        StaticTest s2=new StaticTest();

        s1.setA(10000);
        int res1=s1.getA();

        System.out.println(res1);
        System.out.println(s2.getA());
    }
}
```

예제87)

```
class StaticTest{
    private static int a=10;
    private int b=20;

    public static void printA(){    //정적 메서드에서는 this를 사용하지 못함
        System.out.println(a);
        System.out.println(this.a);    //컴파일 에러 발생
    }

    public void printB(){            //this는 인스턴스 메서드에서 여러 객체에 의해서
        System.out.println(this.b);    //메서드가 호출될 때 이를 구분하기 위해서 사용된다.
    }
}

public class StaticTest03 {
    public static void main(String[] args) {
        StaticTest.printA();
        StaticTest s1 = new StaticTest();
        StaticTest s2 = new StaticTest();
        s1.printB();
        s2.printB();
    }
}
```

예제88)

```
class StaticTest{
    private static int a=10;
    private int b=20;

    public static void printA(){
        System.out.println(a);
        System.out.println(b);    //컴파일 에러 발생
    }

    public void printB(){
        System.out.println(b);
    }
}

public class StaticTest04 {
    public static void main(String[] args) {
        StaticTest.printA();

        StaticTest s1 = new StaticTest();
        StaticTest s2 = new StaticTest();

        s1.printB();
        s2.printB();
    }
}
```

예제89)

```
public class StaticTest05 {
    int b= check(2);
    static int a=check(1);

    public static int check(int i){
        System.out.println("call "+ i);
        return 0;
    }

    public static void main(String[] args) {
        System.out.println("메인");
        StaticTest03 s2=new StaticTest03( );
    }

    static int c=check(3);
}
```


예제90)

```
class StaticTest06 {  
    public static void main(String[] args) {  
        int a=40, b=30, c=10;  
  
        int res;  
        res=Math.max(a, b);  
        System.out.println(a + "와 " + b + " 중 최대값: "+res);  
  
        res=Math.max(b, c);  
        System.out.println(b + "와 " + c + " 중 최대값: "+res);  
    }  
}
```

예제91)

```
class StaticTest07 {  
    public static void main(String[] args) {  
        System.out.println(Math.PI);  
  
        int r=5;  
        double area;  
  
        area=r*r*Math.PI;  
        System.out.println("반지름이 "+r+"인 원의 면적 "+ area);  
    }  
}
```

13. 패키지와 import문

예제92)

```
public class PackageTest01 {  
    public static void main(String[] args) {  
        Random r = new Random();  
  
        for(int i=0; i<10; i++)  
            System.out.println("0 ~ 100 범위의 임의 숫자는 " + r.nextInt(100));  
    }  
}
```

예제93)

```
public class PackageTest02 {  
    public static void main(String[] args) {  
        java.util.Random r = new java.util.Random();  
  
        for(int i=0; i<10; i++)  
            System.out.println("0 ~ 100 범위의 임의 숫자는 " + r.nextInt(100));  
    }  
}
```

예제94)

```
import java.util.*;  
  
public class PackageTest03 {  
    public static void main(String[] args) {  
        Random r = new Random();  
  
        for(int i=0; i<10; i++)  
            System.out.println("0 ~ 100 범위의 임의 숫자는 " + r.nextInt(100));  
    }  
}
```

클래스 정의1)

```
package packTest.packOne;

public class ClassOne {
    public void print(){
        System.out.println("packOne 패키지의 ClassOne 클래스의 print 메서드");
    }
}
```

클래스 정의2)

```
package packTest.packOne;

public class ClassTwo {
    public void print(){
        System.out.println("packOne 패키지의 ClassTwo 클래스의 print 메서드");
    }
}
```

예제95)

```
class PackageTest04
{
    public static void main(String[] args)
    {
        ClassOne object;
        object=new ClassOne();
        object.print();
    }
}
```

예제96)

```
class PackageTest05
{
    public static void main(String[] args)
    {
        packTest.packOne.ClassOne object;
        object=new packTest.packOne.ClassOne();
        object.print();
    }
}
```

예제97)

```
import packTest.packOne.ClassOne;

class PackageTest06
{
    public static void main(String[] args)
    {
        ClassOne object;
        object=new ClassOne();
        object.print();
    }
}
```