



연산자

- 자바에서 활용되는 연산자와 종류를 안다.
- 프로그래밍에서 각각의 연산자에 따라 처리를 효과적으로 할 수 있다.
- 산술 연산자(+, -, *, /, %)의 처리 규칙을 안다.
- 대입 연산자의 흐름 규칙을 알고, 축약형태에 대해서 활용할 수 있다.
- 증감 연산자를 통해서 데이터가 증가/감소하는 **loop**로직을 처리할 수 있다.
- 비교연산자와 논리연산자를 구분하여 사용할 수 있고, 혼합하여 사용할 수 있다.
- **3**항 연산자를 통해서 사용할 수 있는 경우에 대하여 파악하고 활용할 수 있다.

연산란? :

- 연산이란 프로그램에서 데이터를 처리하여 결과를 산출하는 것을 말하고, 이런 연산에 사용되는 표시나 기호를 연산자(operator)

- 산술 연산자

- + :덧셈, - :뺄셈, *:곱셈, /:나눗셈, %:나머지연산

x+y
x-y
x==y
x*y+z

```
int num01=25;
int num02=75;
System.out.println(num01 + " + " + num02 + " = " +(num01+num02));
System.out.println(num01 + " - " + num02 + " = " +(num01-num02));
System.out.println(num01 + " * " + num02 + " = " +(num01*num02));
// 정수/정수 ==> 정수값으로
// 정수/실수 ==> 연산하는 데이터 값 중 하나가 실수값이어야 실수 처리가 됨.
//          (실수형)로 데이터를 cast하여 형변환으로 처리가 가능하다
System.out.println(num01 + " / " + num02 + " = " +(num01/num02));
System.out.println("실수처리"+num01 + " / " + num02 + " = "
                    +(num01/(double)num02));
```



비교연산과 논리연산

- 비교연산 : 두 개의 값의 크기를 비교하는 연산자로 데이터형은 **boolean**으로 저장된다.
 - **a == b, a > b, a < b, a >= b, a <= b, a != b**
 - 주로 if(비교연산), while(비교연산), for(초기:비교연산:증감)에서 활용 된다.
 - ex) if(point >= 80), while(breakChar != 'S')
 - ps) 문자열은 equals를 메서드 활용
 - if(inputID.equals("himan"))
- 논리연산자 : boolean(true/false)을 이용하여 논리곱(&&-and), 논리합(||-or), 배타적 논리합(^-xor), 논리부정(!)로 연산을 수행한다.
 - if(point >= 80 && point < 90) 범위 지정 - **두조건 만족**
 - if(point < 0 || point > 100) - **두조건 중 하나만 만족**



비교 연산자와 논리연산자 예제 :

// 비교연산자

int point = 75;

int passTarget = 60;

System.out.println(point + "==" + passTarget + "?" + (point == passTarget));

System.out.println(point + ">" + passTarget + "?" + (point > passTarget));

System.out.println(point + "<" + passTarget + "?" + (point < passTarget));

System.out.println(point + ">=" + passTarget + "?" + (point >= passTarget));

System.out.println(point + "<=" + passTarget + "?" + (point <= passTarget));

// != 같지 않을 때..

System.out.println(point + "!=" + passTarget + "?" + (point != passTarget));

// 논리연산자

//&&(and) : 두가지 논리조건에 다 합치될 때, true

///**|(or) : 두가지 논리조건 중 하나만이라도 합치될 때, true**

int getPoint = 85;

System.out.println("획득한 점수가 80 이상 90 미만?" +

((getPoint >= 80) && (getPoint < 90)));

System.out.println("점수가 유효하지 않을 조건(0 미만, 100 초과)" +

((getPoint < 0) || (getPoint > 100)));



증감과 대입연산자? :

■ 증감연산자

- 프로그램 작성 시, 특정한 연산자를 통해서 증가하거나 감소시키는 연산자..(1씩 증가 감소)
- $++num$, $num++$, $--num$, $num--$
- `for(int num=0; num<=15; num++)` loop문에 활용

■ 대입연산자

- 변수에 새로운 데이터 값을 할당 처리하는 것을 말한다.
- 변수 = 할당할 데이터; (오른쪽에서 왼쪽 프로세스 흘러감)
- 재귀적 대입연산처리.
 - $num+=3$; $\leftarrow \rightarrow num = num + 3$;
 - $num-=2$; $\leftarrow \rightarrow num = num - 2$;
 - $num *=4$, $num /=5$, $num \% =2$;



증감연산자, 대입연산자 예제 :

```
// 증감 연산자 : 데이터를 1씩 증가 또는 감소시키는 연산자를 말한다.  
int no=1;  
System.out.println("번호(증가)!!!"+(no++));  
System.out.println("번호(증가)!!!"+(no++));  
System.out.println("번호(증가)!!!"+(no++));  
System.out.println("번호(감소)!!!"+(--no));  
System.out.println("번호(감소)!!!"+(--no));  
System.out.println("번호(감소)!!!"+(--no));  
/* 대입연산자 : 변수에 새로운 값을 저장하는 경우에 사용되는 연산자!*/  
int initSaveMoney= 1000;  
// 재귀적 대입연산 : 해당 변수에 덧붙여 데이터를 할당하는 것을 말한다.  
//변수 +=덧붙여할당할 데이터, 변수-=데이터..  
System.out.println("저축!!!! (매월 10만원)");  
int monthCnt=1; // 초기 대입연산처리  
System.out.println( (monthCnt++)+"번째 달!!총 저축금액:"+(  
initSaveMoney+=100000)+"원");  
System.out.println( (monthCnt++)+"번째 달!!총 저축금액:"+(  
initSaveMoney+=100000)+"원");  
System.out.println( (monthCnt++)+"번째 달!!총 저축금액:"+(  
initSaveMoney+=100000)+"원");
```



3항 연산자 :

- 3개의 피연산자를 통하여 결과값을 바로 처리하는 연산 형태를 말한다.
 - 조건식(boolean) ? 조건이 true일때 값: false일때 값
 - [피연산자1] [피연산자2] [피연산자3]
- ex) 나이가 65이상일 때, "경로우대", 그 외일 때, "일반"
 - `int age=67;`
 - `System.out.println("경로우대여부:"+(age>=65?"경로우대":"일반"));`



외부 데이터 값입력..

■ 실행

- java 클래스명[]data01[]data02
- data01, data02는 문자열 배열로 입력되어..
- main(String[] args)에서

args 배열

■ 실행

- c:\Temp\Hello.java를 위치 시킨후에
- command 창을 열어.
- cd ..
- cd ..
- cd Temp
- javac Hello.java ## 컴파일
- java Hello Good day

data01

data02

args 배열

Good

day



Hello.java :

```
public class Hello    // c:\Temp\Hello.java 위치
{
```

args 배열

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

Good

day

```
        System.out.println("Hello World!");
```

```
        System.out.println("첫번째값:"+args[0]);
```

```
        System.out.println("두번째값:"+args[1]);
```

```
    }
```

```
}
```

실습

command 창을 열어.

cd ..(상위dir이동)

cd ..

cd Temp

javac Hello.java ## 컴파일

java Hello Good day

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.

C:\Users\Wacorn>cd ..

C:\Users>cd ..

C:\>cd Temp
```



배열과 list 처리 :

	args[0]	args[1]	args[2]	args[3]	args[4]	args[5]
■ [사과	3000	2	바나나	2000	3

- 배열.length/묶음 단위데이터갯수
- for(int idx=0;idx<args.length/3;idx++)
 - args[0*3] 0 == 과일만 출력..
 - args[1*3] 3
 - args[0*3+1] 1 == 물건값만 출력.
 - args[1*3+1] 4
 - args[idx*3] ➔ 과일명
 - args[idx*3+1] ➔ 물건값
 - args[idx*3+2] ➔ 물건갯수



감사합니다 !
