

printf ("서식문자열", 값)

서식문자를 사용할 수 있는 출력 메소드

- 서식문자 (format 문자) : 따옴표 안에 작성하고 %와 함께한다.

%d : 정수

%f : 실수

%s : 문자열

%c : 문자

System.out.println(10.0 / 3); //3.33333333.. ► 실수의 자리수 조절불가

System.out.printf("%.2f", 10.0/3) //3.33 ► 출력 형식 지정가능

printf("이름: %s", "이순신"); // 이름: 이순신

뒤에 올 값들은 서식문자의 개수와 타입에 맞춰서 따라와야 한다.

마지막에 줄바꿈을 하지 않기 때문에 직접 줄바꿈을 해 줘야 한다.

입력메소드

- 입력: 사용자가 개발자에게 값을 넘겨주는 행위

- 입력 상태

커서가 깜빡이면서 값이 들어오기를 기다리는 상태

값이 입력되지 않으면 다음으로 진행이 불가하다.

- Scanner 란?

화면으로부터 데이터를 입력받는 기능을 제공하는 클래스

```
Scanner sc = new Scanner(System.in);
```

```
int num = sc.nextInt();
```

```
System.out.println(num);
```

. (하위연산자)

A.b = A안의 b = A의 b

next()

통째로가 사용자가 입력한 문자열 값

공백 또는 엔터를 구분점으로 판단하고 분리하여 입력받는다

이 예진: "이", "예진" 분리하기 때문에 "이" 만 입력받는다.

따라서 값의 개수만큼 next()를 사용 해줘야만 한다.

nextLine()

통째로가 사용자가 입력한 문자열 값

공백도 문자로 입력받는다.

앞에 nextLine()을 제외한 다른입력 있다면 의미없는 nextLine()을 하나 미리 써줘야 한다.

nextInt()

통째로가 사용자가 입력한 정수 값

형변환

-강제 형변환

(자료형) : 형변환 연산자

(자료형)값 ex) (double)10 //10.0

-자동 형변환

정수 + 실수 = 실수

ex) 3 + 0.0 = 3.0

문자 + 정수 = 정수

ex) 3 + ' ' = 35 // 'A' + 5 = 70

정수, 실수, 문자 + 문자열 = 문자열

ex) 3 + "Hello" = "3Hello" // 10 + "" = "10" // '3' + "" = "3"

연산자

$$3 * x + 5$$

- 연산자

연산을 수행하는 기호(+, -, *, / 등)

- 피연산자

연산자의 연산 수행 대상

※ 모든 연산자는 연산결과를 반환한다.

- 연산자의 우선순위

하나의 식에 둘 이상의 연산자가 있을 때, 먼저 연산하는 순위를 자동결정 하는것

원하는 우선순위가 있을 경우 괄호()를 사용할 수 있지만, 매번 수동으로 하기 번거로움

최 단 산 관 리 삼 대

최우선 연산자

단항 연산자

산술 연산자

관계 연산자(비교연산자)

논리 연산자

삼항 연산자

대입 연산자

- 결합성

한 문장 안에 우선순위가 같은 여러개 있을 때, 맞는 방향으로 결합하여 연산하는 성질

// 보통 : 왼쪽 ► 오른쪽 ex) $1+2+3+4 \blacktriangleright (1+2)+3+4 \blacktriangleright (3+3)+4 \blacktriangleright 6+4 \blacktriangleright 10$

※ 예외

- 대입연산자, 단항연산자는 오른쪽 > 왼쪽

ex) $x = y = 5 \blacktriangleright x = 5 \blacktriangleright 5$

최우선 연산자

()

단항 연산자(피연산자가 하나)

형변환 연산자: (자료형)값

- : 피연산자의 부호를 반대로 변경

```
int num= -10; ▶ num = -num; // -(-10) ▶ sysout(num) //10
```

산술 연산자

+, -, *, / (사칙연산)

% (모듈러스) : a%b // a를 b로 나눈 나머지 ex) 10 % 3 // 1

관계 연산자(비교연산자)

두 피연산자를 비교해서 true 또는 false 반환

>, < : 초과, 미만

>=, <=: 이상, 이하

== : 같다

!= : 다르다

ex) 4 < 16 // true

- 조건식

조건을 제시하는 식 / 결과가 참 또는 거짓(true, false) 둘 중 하나로 나오는 식

관계연산자를 이용해서 만들어지는 식이다.

자바에서는 0 과 1 대신 false, true 라는 값을 사용한다.

논리형 boolean true, false

논리 연산자

조건식을 **연결**할 때 사용하는 연산자

* A 와 B는 조건식 가정

A && B(AND결합) : 두 조건식의 연산결과가 둘다 true 라면 true
둘 중에 하나라도 false 라면 false

A || B(OR결합) : 두 조건식의 연산 결과중 하나라도 true면 true

ex) $y > 5 \ \&\& \ y < 10$

관계연산자를 먼저 수행된 후, 논리연산자가 수행

$5 < y < 10$ // 에러

삼항 연산자

조건식 ? 값1(참이면 오는 부분) : 값2(거짓이면 오는 부분)

조건식의 연산결과가 **true**라면 값1 선택

조건식의 연산결과가 **false**라면 값2 선택

삼항연산자는 **사용부분 통째로**가 하나의 값

경우의 수가 **딱 두가지의 값**으로 나오는 경우에 사용하는 것이 좋다.

대입 연산자(=)

연산이 모두 끝나야 연산결과를 저장할 수 있기 때문에 **우선순위가 가장 낮다**.

아스키코드

A ~ Z (대문자) 65 ~ 90

십진법	모양	십진법	모양
65	A	78	N
66	B	79	O
67	C	80	P
68	D	81	Q
69	E	82	R
70	F	83	S
71	G	84	T
72	H	85	U
73	I	86	V
74	J	87	W
75	K	88	X
76	L	89	Y
77	M	90	Z

a ~ z (소문자) 97 ~ 122

십진법	모양	십진법	모양
97	a	110	n
98	b	111	o
99	c	112	p
100	d	113	q
101	e	114	r
102	f	115	s
103	g	116	t
104	h	117	u
105	i	118	v
106	j	119	x
107	k	120	x
108	l	121	y
109	m	122	z

0 ~ 9 (숫자) 48 ~ 57

십진법	모양	십진법	모양
48	0	53	5
49	1	54	6
50	2	55	7
51	3	56	8
52	4	57	9

※ 필수로 외우기

숫자 0 = 48

숫자 9 = 57

대문자 A = 65

대문자 Z = 90

소문자 a = 97

소문자 z = 122