* 전위 연산자 (+, -, ++, --)

부호 예) -a, +a

```
int a = 100;
++a;
```

컴파일 할 때 변환 → a = a + 1; (a = 100) → bytecode

a? ⇒ 101

```
int a = 100;
--a;
```

→ a = a - 1; (a = 100)

a? ⇒ 99

assignment operator (할당연산자, 배정연산자) 대입연산자

a = a + 1;

l-value = left-value ⇓ 메모리

right-value = r-value ⇓ 값

a = 100 + 1;

a = 101;

* 주의!
항상 r-value가 모두 실행된 후
마지막으로 '=' 연산자가 실행된다.

* 후위 연산자 (++, --)

```
int a = 100;
a++;
```

↳ { int temp = a;
  a = a+1; }

a? ⇒ 101

```
int a = 100;
a--;
```

↳ int temp = a;
a = a-1;

a? ⇒ 99

```
int a =100;
a = a++;
```

→ { int temp = a;
  a = a+1;
  a = temp; }

a? 100!

```
int a = 100;
int b;
b = ++a;
```

컴파일 시 | 변환

a = a+1;
b = a;

a? 101
b? 101

```
int a= 100;
int b;
b = a++;
```

컴파일 시 | 변환

int temp = a;
a = a+1;
b = temp;

a? 101
b? 100

```
int i = 100;
System.out.println(i++);
```

② println ① i++

메서드 호출 전에 파라미터로 주어진 표현식을 먼저 실행한다

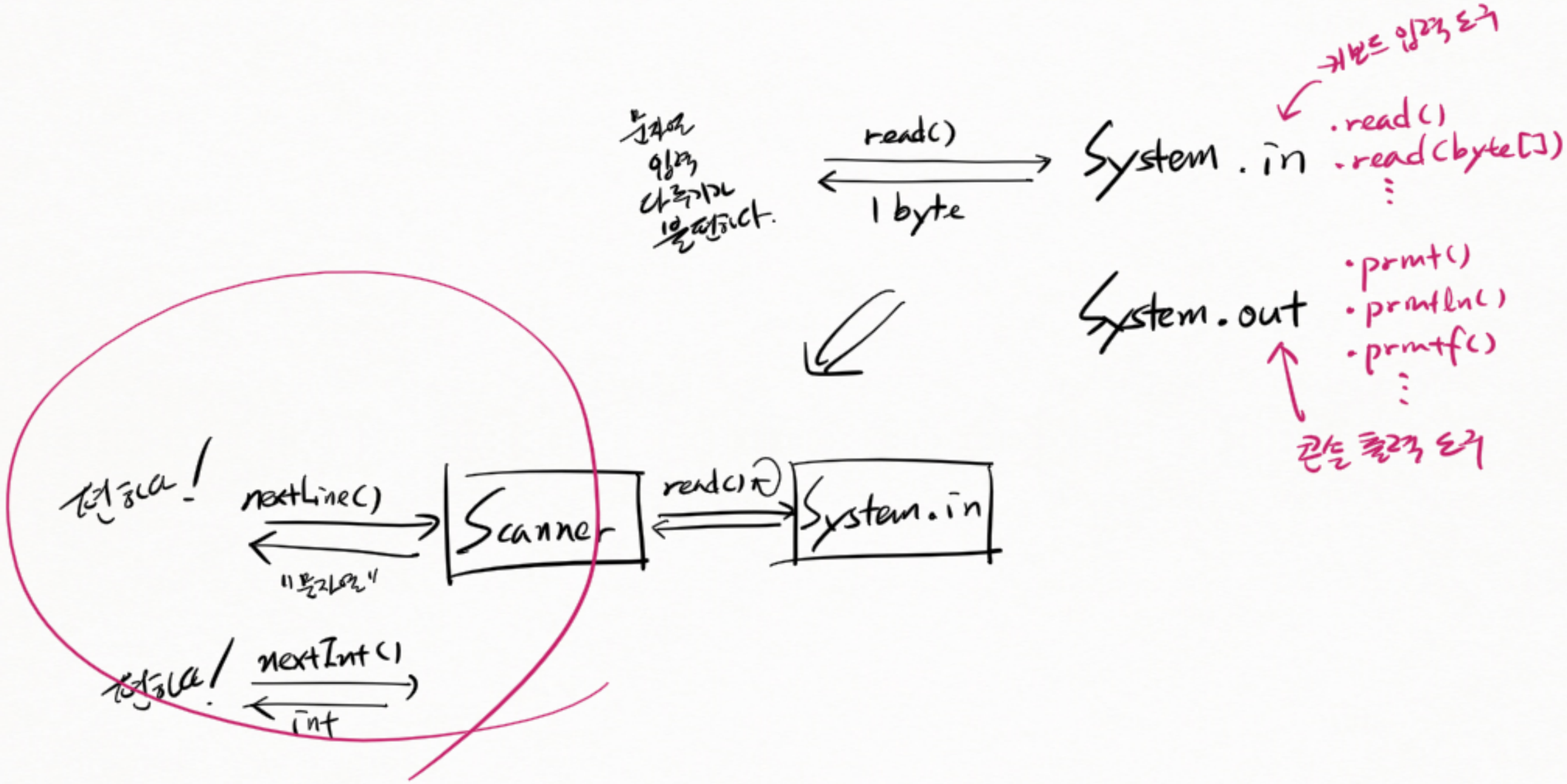
→

```
{
int temp = i;
i = i+1;
System.out.println(temp);
```

```
int x = 100;   (101)
(++x)++;
 101
```

```
int i = 2, 3, 4, 5
int result = i++ + i++ * i++;
             2      3      4
             2  +   12
                14
r = 14
```

* 키보드 입력 다루기


키보드 입력 도구
.read()
.read(byte[])
read()
1 byte
System.in
.prnt()
.prntln()
.prntf()
System.out
콘솔 출력 도구
전달해줘!
nextLine()
"문자열"
Scanner
read()
System.in
전달해줘!
nextInt()
int

* 키보드 입력 다루기 : nextInt(), nextFloat(), nextBoolean()

사용자
↓ 입력
100 3.14 true⏎

"100" → nextInt() → return → 100 / int

"3.14" → nextFloat() → return → 3.14 / float

"true" → nextBoolean() → return → true / boolean

nextInt() → nextFloat() → nextBoolean()

"통행허가코드"

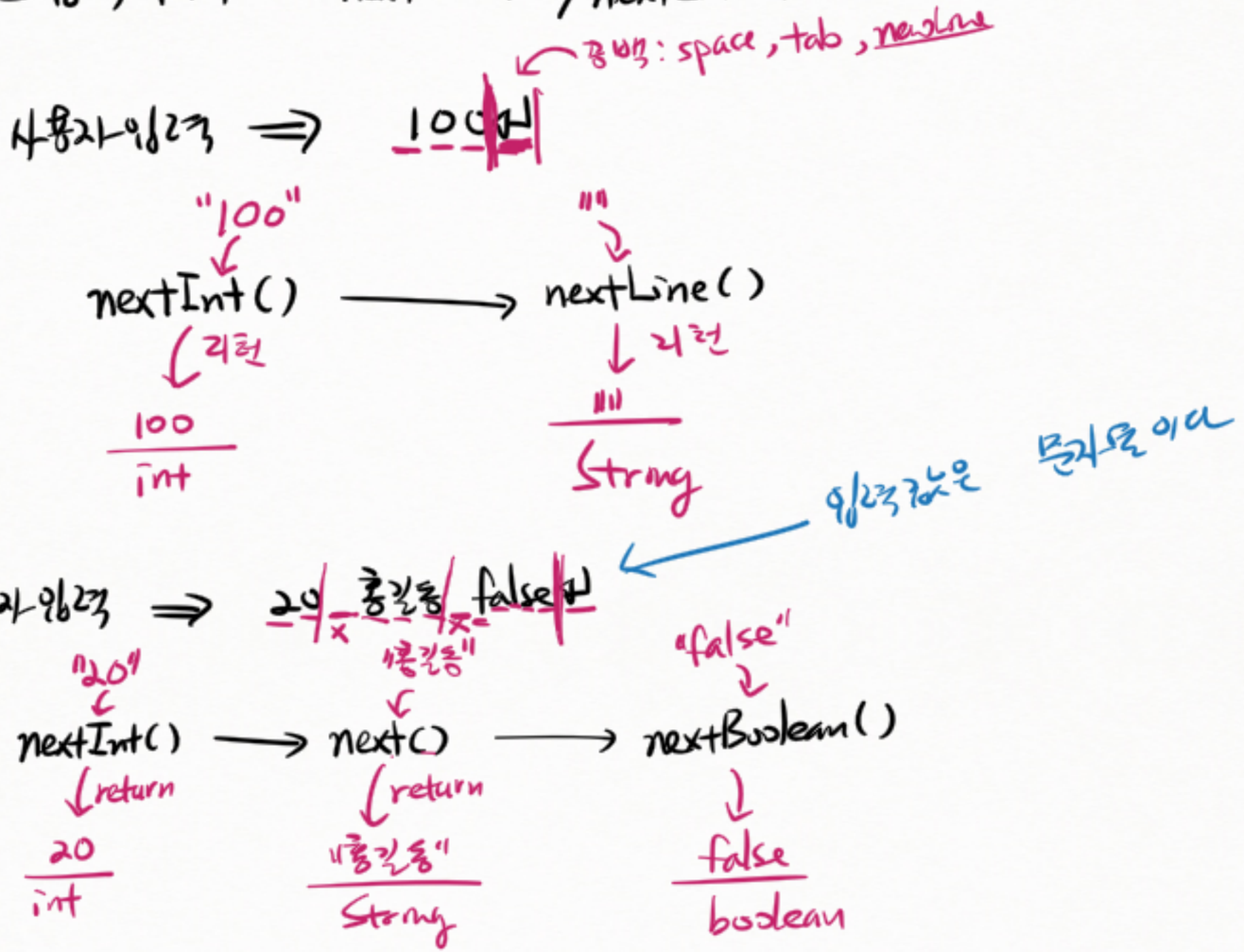
token?
- 공백으로 구분되는 단어
↓
whitespace
- 스페이스
- 탭
- 줄바꿈

예) 100 3.14 true
(각각 token)

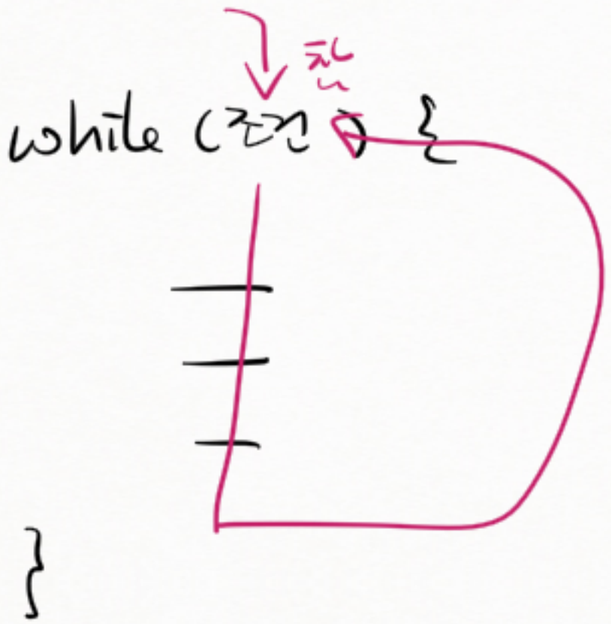
상황 → context

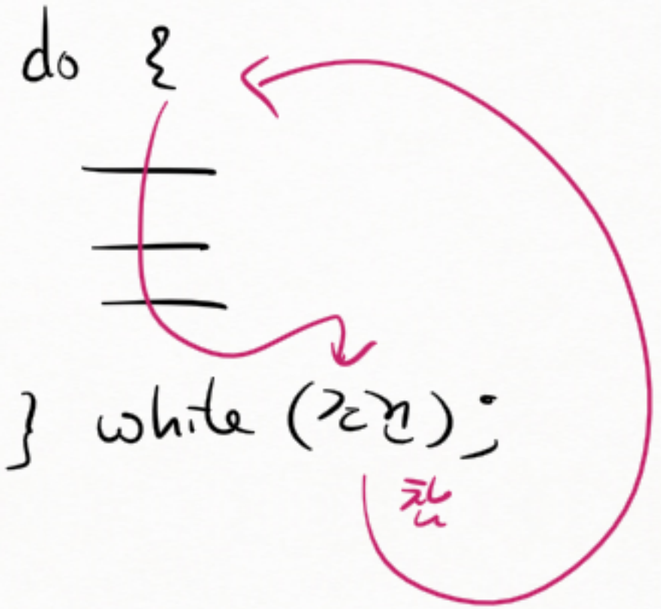
상황인지시스템
Context Intelligence

* 키보드 입력 받기 - nextInt() / nextLine()



* while , do ~ while





* for(; ;)