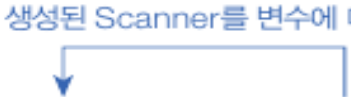


키보드 입력 데이터를
변수에 저장하는 방법

▣ 키보드 입력 데이터를 변수에 저장


생성된 Scanner를 변수에 대입

<스텝01> Scanner scanner = new Scanner(System.in);
scanner 변수 선언 Scanner 객체 생성

A blue arrow originates from the text '생성된 Scanner를 변수에 대입' and points down to the 'new Scanner(System.in);' part of the code line in Step 01.

읽은 문자열을 String 변수에 저장

<스텝02> String inputData = scanner.nextLine();
String 변수 선언 (Enter) 키를 누르면 입력된 문자열을 읽음

A blue arrow originates from the text '읽은 문자열을 String 변수에 저장' and points down to the 'scanner.nextLine();' part of the code line in Step 02.

<스텝03> scanner.close();

● Scanner 주요 메소드

- Scanner 모든 메소드는 [Enter]시 입력한 값 끝에는 자동으로 Wn 이 추가되어 집니다.(Enter 와 Wn는 같은 의미)

- ◇ nextLine()은 개행 문자(Wn)을 포함하는 한 라인을 읽고,
버퍼에 입력된 문자열에서 개행 문자(Wn)까지 다 가져온다. => 버퍼에 개행 문자(Wn) 남아 있지 않음
- ◇ Scanner 나머지 메소드는 공백(토큰)을 기준으로 읽고,
버퍼에 입력된 문자열에서 개행 문자(Wn)를 가져오지 않는다. => 버퍼에 개행 문자(Wn) 남아 있음

메소드	설명
String next()	다음 토큰을 문자열로 리턴
byte nextByte()	다음 토큰을 byte 타입으로 리턴
short nextShort()	다음 토큰을 short 타입으로 리턴
int nextInt()	다음 토큰을 int 타입으로 리턴
long nextLong()	다음 토큰을 long 타입으로 리턴
float nextFloat()	다음 토큰을 float 타입으로 리턴
double nextDouble()	다음 토큰을 double 타입으로 리턴
String nextLine()	'\n'을 포함하는 한 라인을 읽고 '\n'을 버린 나머지만 리턴
void close()	Scanner의 사용 종료
boolean hasNext()	현재 입력된 토큰이 있으면 true, 아니면 새로운 입력이 들어올 때까지 무한정 기다려서, 새로운 입력이 들어오면 그 때 true 리턴. ctrl-z 키가 입력되면 입력 끝이므로 false 리턴

● next() 와 nextLine() 차이점 : 개행문자(Wn)를 무시하냐 안하냐의 차이

▷ next()

- 공백을 기준으로 한 단어 또는 한 문자씩 입력받는다.
- 버퍼에 입력된 문자나 문자열에서 공백 전까지의 단어를 읽는다.
- 개행 문자를 가져오지 않는다.

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        String str = scan.next();  
        System.out.println(str);  
    }  
}
```

```
Hello Wrold  
Hello
```

▷ nextLine()

- 문자 또는 엔터를 치기 전까지의 문장 전체를 입력받는다.
- 버퍼에 입력된 문자열을 개행 문자까지 다 가져온다.

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scan = new Scanner(System.in);  
        String str = scan.nextLine();  
        System.out.println(str);  
    }  
}
```

```
Hello World  
Hello World
```

● 입출력시 주의사항

- nextLine() 만 사용해서 입력받고, 변수 타입에 맞게 타입 변환후 저장하는 한다.

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        String name, strAge;  
        int age;  
  
        System.out.println("당신의 이름을 입력하세요.");  
        name = scanner.nextLine();  
  
        System.out.println("당신의 나이를 입력하세요.");  
        strAge = scanner.nextLine();  
        age = Integer.parseInt(strAge);  
  
        System.out.printf("당신의 이름은 %s입니다.\n",name);  
        System.out.printf("당신의 나이는 %d세 입니다.\n",age);  
    }  
}
```

당신의 이름을 입력하세요.

홍길동

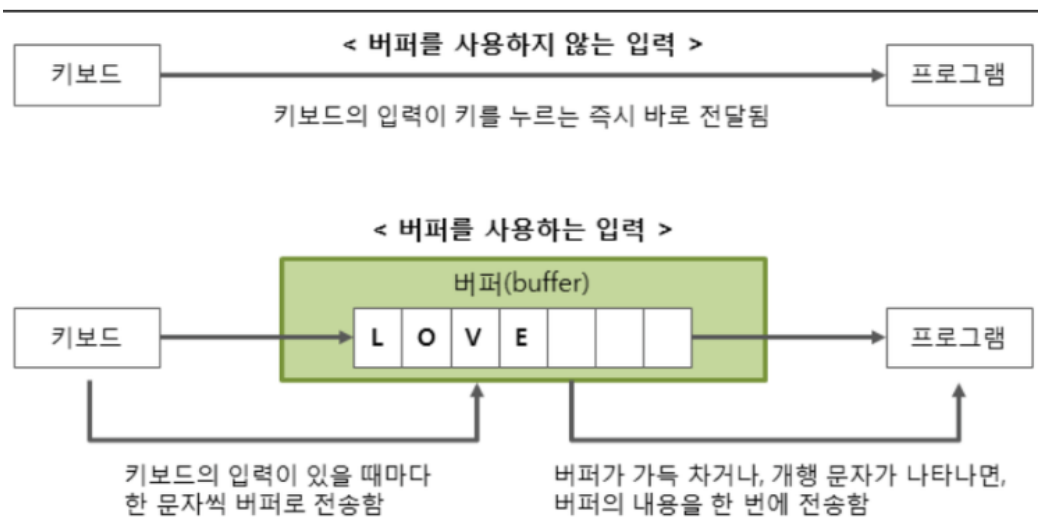
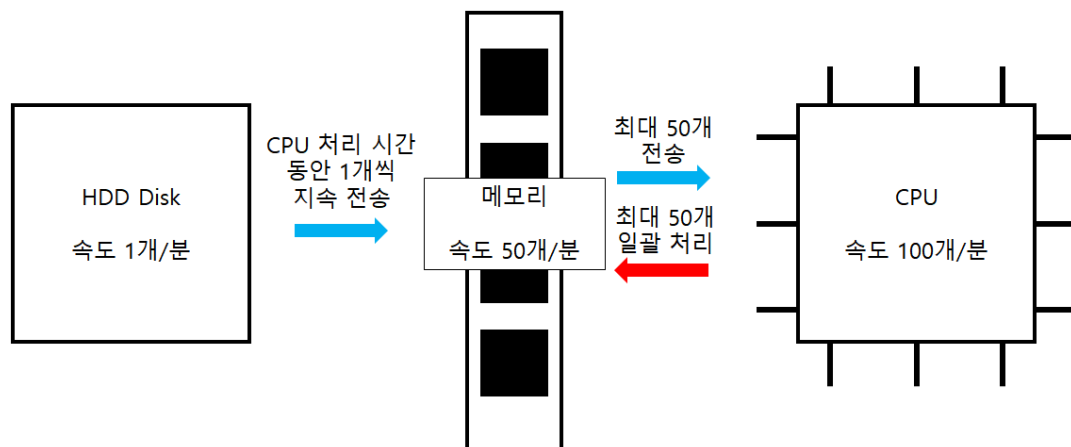
당신의 나이를 입력하세요.

10

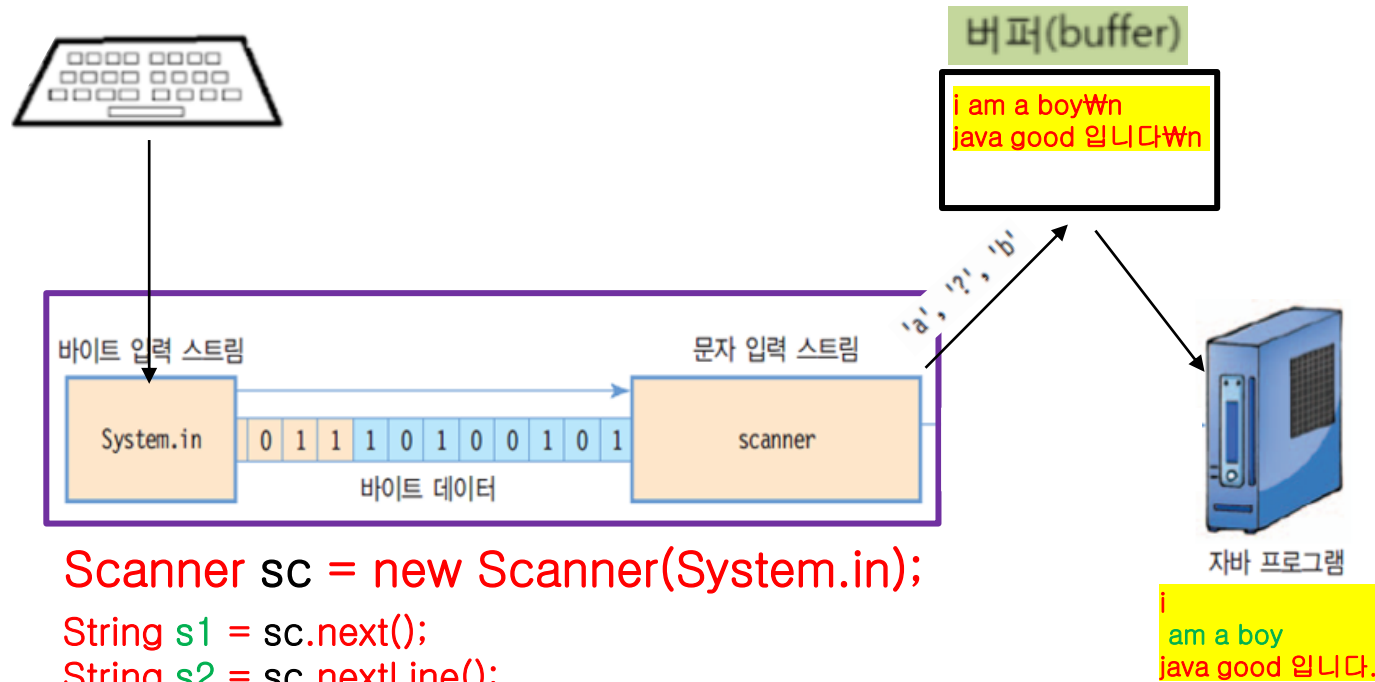
당신의 이름은 홍길동입니다.

당신의 나이는 10세 입니다.

※ 버퍼(buffer)



- Scanner의 next 등의 메소드는 버퍼가 비어 있으면 대기, 데이터가 존재하면 데이터를 가져옵니다.



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

```
String s1 = sc.next();
```

```
String s2 = sc.nextLine();
```

```
String s3 = sc.nextLine();
```

```
System.out.println(s1);
```

```
System.out.println(s2);
```

```
System.out.println(s3);
```