기본 자료형

자바에는 '기본 자료형(Primitive Types)'이 있습니다. 변수 단원에서 배운 자료형 중에 가장 기본이 되는 자료형들입니다.

Type	Bits	Range of Values
byte	8bits	-2^7 ~ 2^7-1 (-128 ~ 127)
short	16bits	-2^15 ~ 2^15-1 (-32768 ~ 32767)
int	32bits	-2^31 ~ 2^31-1 (-2147483648 ~ 2147483647)
long	64bits	-2^63 ~ 2^63-1 (-9223372036854775808 ~ 9223372036854775807)
float	32bits	*single-precision 32-bit IEEE 754 floating point
double	64bits	*double-precision 64-bit IEEE 754 floating point
char	16bits	₩u0000 ~ ₩uffff (0 ~ 2^15-1)
boolean	*VMD	true, false

각 자료형의 기본 값은 다음과 같습니다.

type	기본 값
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'₩u0000'
boolean	false

숫자형

숫자를 담는 자료형은 정수형 **byte**, **short**, **int**, **long**, 그리고 소수형 **float** 과 **double** 이 있습니다.

정수형

자바에서는 **int** 를 정수형의 메인으로 사용합니다. 즉, 정수를 입력하면 기본적으로 **int** 로 간주합니다.

```
byte a = 127;
```

```
byte a = 127;
```

하지만 이렇게 **127** 을 **byte** 변수에 지정해주면, **127** 은 **byte** 의 범위 내에 있기 때문에 컴파일러가 알아서 **127** 을 **byte** 로 바꿔줍니다.

```
long x = 12345678910;  // 오류 (정수 범위를 벗어남)
```

만약 **int** 의 범위에서 벗어나는 값을 써주면 오류가 나옵니다. 고치기 위해서는 **L** 을 붙여주면 됩니다.

long 을 표현하기위한 리터럴은 뒤에 **l** (소문자 엘)또는 **L** (대문자 엘)을 붙여줍니다. 소문자 **l** (엘)의 경우 숫자 **1** (일)과 혼동될 수 있으니 대문자 **L** (엘)을 적는 것이 권장사항입니다.

```
long x = 12345678910l; // 소문자 l은 1과 헷갈릴 수 있음

long x = 12345678910L; // 권장사항
```

소수형

자바에는 두 가지 소수형 자료형이 있습니다: **float** 과 **double** .

float 과 **double** 은 둘다 소수형을 담지만 정밀도(Precision) 혹은 정확도에 차이가 있습니다. **double** 이 더 정밀하게 값을 보관할 수 있습니다. 이러한 이유로 자바에서는 **double** 을 소수형의 기본으로 사용합니다. 정수를 입력하면 기본적으로 **int** 로 인식되듯이, 소수를 입력하면 기본적으로 **double** 로 인식되는 것이죠.

만약 **float** 를 쓰고 싶으면 소수를 쓰고, 뒤에 **f** 를 붙여주면 됩니다.

```
float f = 3.14f;
```

글자

```
char a1 = 'a';
char a2 = 97;
char a3 = '\u0061';
char a4 = '가';
```

글자(Character) 하나를 담는 자료형 **char** 도 있습니다. **char** 은 딱 글자 하나만 넣어줄 수 있고, 작은 따옴표로 글자를 둘러싸야 합니다. 글자 여러 개를 담고 싶으면 큰 따옴표로 둘러싸고 아래에서 배울 **String** 이라는 자료형을 사용하면 됩니다.

a1 에는 글자 **'a'** 를 담았고, **a2** 에는 글자 **'a'** 에 해당하는 ASCII (https://ko.wikipedia.org/wiki/%EB%AF%B8%EA%B5%AD%EC%A0%95%EB%B3%B4%EA%B5%90%ED%99%9 값인 **97** 을 담았습니다. 사실 **a1** 과 **a2** 는 같은 셈이죠.

a3 는 유니코드 (https://ko.wikipedia.org/wiki/%EC%9C%A0%EB%8B%88%EC%BD%94%EB%93%9C) 값입니다. **'가'** 도 유니코드 중 하나라 자바의 **char** 에 담을 수 있습니다.

ASCII 코드와 유니코드에 대해서 궁금하시면 더 찾아보시길 바랍니다!

불린

```
boolean myBoolean = true;
myBoolean = false;
```

참(true), 거짓(false)을 담는 boolean 입니다. 값으로는 **`true`** 와 **`false`** 가 가능합니다.

나중에 '제어문' 섹션에서 보시면 용도를 더 쉽게 이해하실 수 있을 것입니다.

문자열

이제 조금 특별한 자료형을 살펴볼까요? **`String` 은 기본 자료형이 아닙니다.** **`String`** 은 클래스입니다. 클래스를 변수의 형으로 쓰고 변수를 선언하면 그 변수는 클래스의 인스턴스를 담을 수 있습니다. '객체 지향 프로그래밍' 섹션에서 제대로 배우니까, 일단은 이해가 안 되도 그냥 넘어가시면 됩니다.

먼저 **`String`** 은 큰 따옴표(**`"`**)로 둘러싸인 글자들을 적어 만들 수 있습니다.

```
String a = "Hello, I'm ";
String b = ".";

System.out.print(a);
System.out.print(26);
System.out.println(b);


Hello, I'm 26.
```

다양한 활용법은 차차 알아보도록 합시다!

☑ 수업을 완료하셨으면 체크해주세요.

수강생 Q&A 보기 (/questions?assignment_id=261&sort_by=popular)

질문하기 (/questions/new?assignment_id=261&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

이전 강의 자료형 (/assignments/251)

다음 강의 자기소개서 만들기 (/assignments/269)