

# 자바

## 1. 변수

메모리상에서 **값을 저장할 수 있는 공간**을 의미한다(이는 저장된 값이 변경될 수 있어 수학적 용어에서 유래했다). 하나의 변수는 **단 하나의 값만 저장**할 수 있으며, 새로운 값을 저장하면 기존의 값은 사라진다.

### 변수 사용

변수를 사용하기 위해서는 먼저 변수를 선언해야함.

ex) int age

- int → 변수 타입 - age → 변수 이름.

- 변수 타입은 변수에 저장될 값이 어떤 '타입'인지를 지정하는 것이다. 저장하고자 하는 값의 종류에 맞게 변수의 타입을 선택해서 적어주면 된다.

↳ (자바에는 정수형, 실수형, 문자형 등 다양한 타입을 제공)

- 변수 이름은 말 그대로 변수의 이름이다. 값을 저장할 수 있는 메모리공간(변수)에 이름을 지어주는 것.

↳ (같은 이름의 변수가 여러 개 존재해서는 안된다. 서로 구별 할 수 있어야되기 때문.)

### 변수의 초기화

변수를 선언한 이후부터는 변수를 사용할 수 있으나, 그 전에 반드시 변수를 '**초기화**'해야한다.

메모리는 여러 프로그램이 공유하는 자원이므로 전에 다른 프로그램에 의해 저장된 '알 수 없는 값'이 남아있을 수 있기 때문이다.

👉 (변수의 초기화란, 변수를 사용하기 전에 처음으로 값을 저장하는 것)

### 변수의 명명규칙

프로그래밍에서 사용하는 모든 이름을 '**식별자**'라고 하며, 식별자는 같은 영역 내에서 서로 구분(식별)될 수 있어야 한다.

그리고 식별자를 만들 때는 다음 규칙을 지켜야한다.

1. 대소문자가 구분되며 길이에 제한이 없다. ex) Java와 java는 다른 것으로 간주된다.

2. 예약어를 사용해서는 안된다. ex) true는 예약어라서 사용이 불가능 하지

만 True는 가능.

3. 숫자로 시작해서는 안된다. ex) top10은 가능하지만 10up은 불가능.

4. 특수문자는 '\_', '\$'만을 허용한다. \$sharp은 허용되지만 S#arp는 불가능.

필수는 아니지만 암묵적인 룰

1. 클래스 이름의 첫 글자는 항상 대문자로 한다. (변수와 메서드의 이름의 첫글자는 항상 소문자로 한다.)

2. 여러 단어로 이루어진 이름은 단어의 첫 글자를 대문자로 한다.(lastIndexOf, StringBuffer)

3. 상수의 이름은 모두 대문자로 한다. 여러 단어로 이루어진 경우 '\_'로 구분한다. (PI, MAX\_NUMBER)

## 1. 변수의 타입

우리가 주로 사용하는 값의 종류는 크게 '문자와 숫자'로 나눌 수 있으며, 숫자는 다시 '정수와 실수'로 나눌 수 있다.

이러한 값의 종류에 따라 값이 저장될 공간의 크기와 저장형식을 정의한 것이 '

**자료형**'이다.

자료형에는

**문자형(char)**, **정수형(byte, short, int, long)**, **실수형(float, double)** 등이 있으며,

변수를 선언할 때는 저장하려는 값의 특성을 고려하여 가장 알맞은 자료형을 변수의 타입으로 선택하면 된다.

### 기본형과 참조형

자료형은 크게 '**기본형**'과 '**참조형**'으로 나눌 수 있는데,

기본형 변수는 '

**실제 값을 저장**' 하는 반면,

참조형 변수는 어떤 값이 저장되어있는 '

**주소를 값으로 갖는다.**'

#### 기본형

- 논리형(boolean), 문자형(char), 정수형(int, short, byte, long), 실수형(float, double)

계산을 위한 실제 값을 저장한다. 모두 8개

#### 참조형

- 객체의 주소를 저장한다. 8개의 기본형을 제외한 나머지 타입.

참조형 변수를 선언할 때는 변수의 타입으로 클래스의 이름을 사용하므로 클래스의 이름이 참조변수의 타입이 된다.



### 참조변수 선언 방법

클래스이름 변수이름; //변수의 타입이 기본형이 아닌 것들은 모두 참조변수이다.

```
Date Today = new Date();
```

Date클래스 타입의 참조변수 today를 선언한 것이다.

참조변수는 null 또는 객체의 주소를 값으로 갖으며 참조변수의 초기화는 다음과 같다.

객체를 생성하는 연산자 new의 결과는 생성된 객체의 주소이다.

이제 참조변수 today를 통해서 생성된 객체를 사용할 수 있게 된다.

### 자료형과 타입의 차이점

기본형은 저장할 값의 종류에 따라 구분되므로 기본형의 종류를 얘기할때는 '**자료형**'이라는 용어를 사용

참조형은 '

**객체의 주소**'를 저장하므로 값이 아닌, 객체의 종류에 의해 구분되므로 참조형 변수의 종류를 구분할 때는 '**타입**'이라는 용어를 사용.

타입이 자료형을 포함하는 보다 넓은 의미의 용어이므로 굳이 구분하지 않아도 됨.

## 2.1 기본형

기본형에는 모두 8개의 타입(자료형)이 있으며, 크게 논리형, 문자형, 정수형, 실수형으로 구분된다.

**논리형(boolean)** - true와 false 중 하나를 값으로 갖으며, 조건식과 논리적 계산에 사용된다.

**문자형(char)** - 문자를 저장하는데 사용되며, 변수에 하나의 문자만 저장할 수 있다.

**정수형(byte, short, int, long)** - 정수를 저장하는데 사용되며, 주로 int가 사용된다. byte는 이진 데이터를 다룰 때 사용되며, short은 C언어와의 호환을 위해서 추가되었다.

**실수형(float, double)** - 실수를 저장하는데 사용되며, 주로 double이 사용된다.

| boolean을 제외한 7개의 기본형은 서로 연산과 변환이 가능하다.



	1byte	2byte	4byte	8byte
논리형	boolean			
문자형		char		
정수형	byte	short	int	float
실수형			float	double



**boolean**은 true와 false 두 가지 값만 표현할 수 있으므로 가장 작은 크기인 1byte.

**char**은 자바에서 유니코드(2byte 문자체계)를 사용하므로 2byte.

**byte**는 크기가 1byte라서 byte

**int**(4byte)를 기준으로 짧아서 **short**(2byte), 길어서 **long**(8byte).

**float**는 실수값을 부동소수점(floating-point) 방식으로 저장하기 때문에 float.

**double**은 float보다 두 배의 크기를 갖기 때문에 double.

정수형이 가질 수 있는 값의 범위 ->  $-2^{n-1} \sim 2^{n-1} - 1$  (n은 비트 수)



7~9자리의 수를 계산할 때는 넉넉하게 long타입(약 19자리)으로 변수를 선언하는 것이 좋음.

연산중에 저장범위를 넘어서게 되면 원하지 않는 값을 결과로 얻게 될 것이기 때문

## 2.2 상수와 리터럴

상수는 변수와 마찬가지로 '값을 저장하는 공간'이지만, 변수와 달리 한 번 값을 저장하면 다른 값으로 변경할 수 없다

선언방법은 변수 선언과 같으며 앞에 **final**을 붙여주기만 하면 된다.

상수는 반드시 '선언과 동시에 초기화'해야 하며, 그 후 부터는 상수의 값을 변경하는 것이 허용되지 않는다.

상수의 이름은 모두 대문자로 하는 것이 암묵적인 관례이며, 여러 단어로 이루어져있는 경우 '\_'로 구분한다.