



# (6.15)2일차

강의 번호	1주차
복습	<input checked="" type="checkbox"/>
유형	변수
자료	
작성일시	@2021년 6월 15일 오후 3:36

## 1. 프로그래밍

### 1) 프로그램 : 명령어가 나열됨

→ 컴퓨터가 알아들을 수 있는 것

(ex)시간대별로 프로그램이 배치되어있는것...

### 2) 프로그래밍(코딩) : 프로그램을 만드는 자체

### 3) 프로그래머 : 프로그램을 만드는(코딩) 사람들

## 2. 자바

- 자바의 아버지 : 제임스 고슬링!! 알아두기

### 1) 운영체제에 독립적 : 운영체제에 신경 안씀!

(윈도우에서 만든 것->맥북에서도 할 수 있음!, 물론 맥에서 윈도우로 돌릴수도 있음)

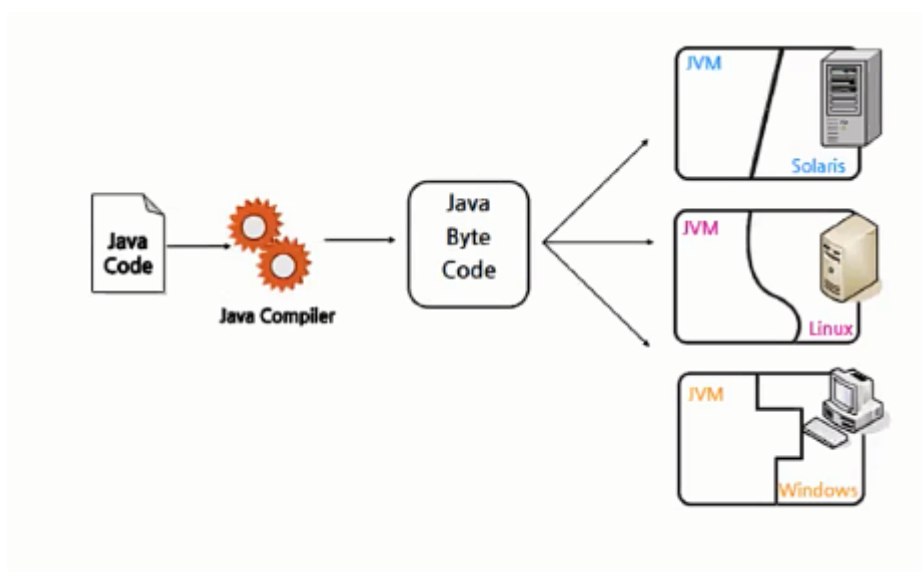
### 2) 사용하기 쉬운 언어

- 다른 언어의 단점 보완(포인터, 메모리 관리)  
→ 프로그래밍을 할때 메모리관리가 정말 중요해...근데 자바하면 신경쓸필요x
- 객체 지향언어
- 능률적이고 명확한 코드 작성 가능

- 3) 자동 메모리 관리 : 사용하지 않는 메모리가 있다면 자동으로 없애버리기 때문에 정말 유용함
- 4) 네트워크와 분산환경 지원
- 5) 멀티쓰래드 지원 : 여러사람이 접근해서 하나의 프로그램을 만들기 쉽게 함

### 3. JVM

#### 1) 설치범위



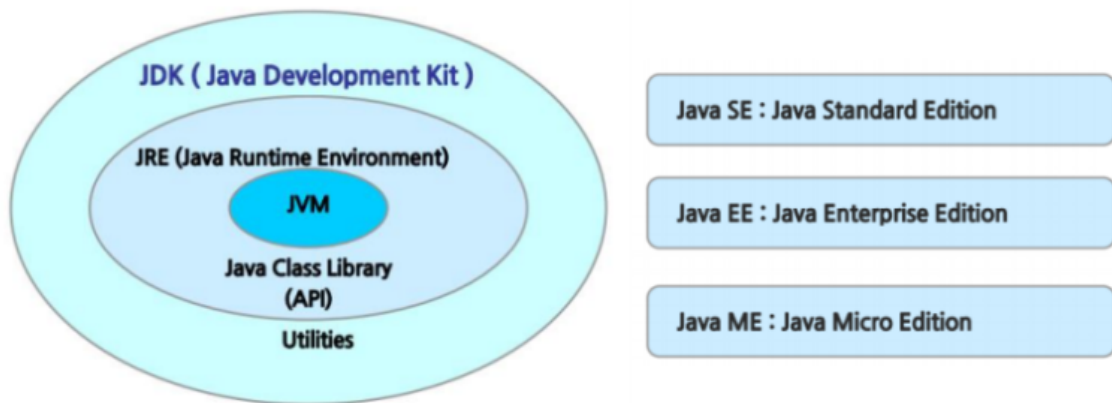
**Java code** (사람이 사용하는 언어: .java로 입력됨)

→ **java compiler**(컴퓨터가 알아들을 수 있는 언어로 해석해줌 )

→ 해석결과 **Java byte code**(.class파일)로 나옴

→ **JVM이** .class 파일을 한번 더 해석해서 실행됨

#### 2) 설치범위(그림 잘 기억하기!)



- jvm : 무조건 있어야 작성되어 있는 코드를 실행할수 있음
- jre : 실행시 필요한게 들어있는거  
(개발안하는 사람이 강 실행시키는 것...)
- jdk : 개발자는 자바를 가지고 개발을 해야 하므로 필요!!  
이거만 설치하면 다 끝남!

#### \*프롬프트 관리자

- java -version : 현재 내가 가지고 있는 자바버전
- javac -version : 현재 내가 가지고 있는 자바버전

\*amazon coretto : 환경변수 설정 안해도 자동으로 깔아줌...!

---

//3교시 😎

[변수]

#### 1. 정의

- 메모리 공간(RAM)에 한 개의 값을 기록하기 위한 장소(공간)

→ 크기가 한정적이기 때문에, 이클립스(자바)가 사용할수 있는 공간 조차도 한정적이다.

그러므로 우리는 공간안에 저장하고 싶은 장소(방)의 크기도 내 멋대로 조절할 수 없다.

종류에 따라 그 크기가 결정되 있다.(원룸, 아파트, 오피스텔...)

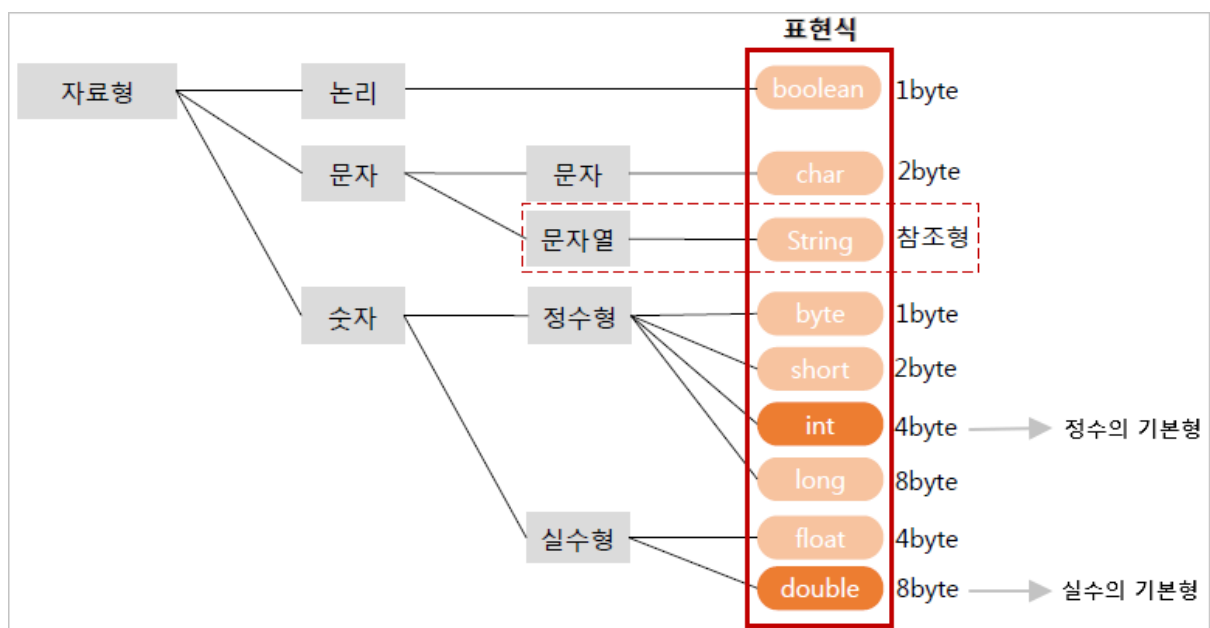
## 2. 변수의 자료형(Type)

### 1) 기본형(Primitive Type)

- 실제 데이터 값을 저장 (내가 저장하고 싶은게 8이면 8그대로 저장)
- 논리형, 문자형, 정수형, 실수형으로 나뉘어지고 8개의 자료형이 있음
- 각 자료형 별 데이터 저장 크기가 다름

### 2) 참조형(Reference Type)

- 방의 크기가 타입별로 다르다.
- 데이터가 저장되어 있는 주소의 정보를 저장  
(8을 다른 공간(b)에 저장하고, b라는 공간을 x100이라는 주소값을 줌  
그리고 나서 원래 부터 있던 a라는 공간에 x100이라고 지칭했던 주소값을 집어넣음)
- 기본형을 제외한 나머지(String 등), 사용자 정의 자료형
- 4Byte의 공간을 저장공간으로 할당 : 주소만 들어가면 되서 only 4byte만 제공



boolean, char, string, int, long, double을 많이 사용함 /어떤 타입이 있는지, 타입별 크기는? 암기

- 표현식 : 자바에서 방을 어떤 타입인지 지칭할때 '논리형'타입이다 (x)  
'boolean'타입이다 (o)
- 해당하는 타입의 데이터만 들어갈수 있다...  
(ex\_□모양의 값이 별표안에 들어갈수 없음)

### 3. 변수 저장 가능 범위(외울필요는 없음...)

자료형	범 위	크기(bit)
boolean	true, false	8
char	0~65,535(유니코드문자)	16
byte	-128 ~ 127	8
short	-32,768 ~ 32,767	16
int	-2,147,483,648 ~ 2,147,483,647	32
long	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	64
float	$\pm 1.4E-45 \sim 3.4E38$	32
double	$\pm 4.9E-324 \sim 1.8E308$	64

컴퓨터는 2진수로 인지하기 때문에  $2^n$  ( $n$  = 비트크기)로 범위가 할당됨

boolean은 true 와 false로 저장된다...

### 4. 변수의 선언

#### 1) 메모리 공간에 데이터를 저장할 수 있는 공간을 할당

→ 메모리 안에 특정한 공간을 만들겠다. 근데 그 공간의 타입과 이름이 무엇이나?를 정하는 것 까지 변수의 선언!

**자료형**

변수타입지정

**변수명**

변수명지정

**;** 마침

✓ 선언 예시

<pre>// 논리형 변수 선언 boolean isTrue;  // 문자형 변수 선언 char ch;  //문자열 변수 선언 String str;</pre>	<pre>// 정수형 변수 선언 byte bnum; short snum; int inum; long lnum;  // 실수형 변수 선언 float fnum; double dnum;</pre>
---	--

## 5. 변수 명명 규칙

(1) 대소문자가 구분되며 길이 제한이 없다

- 대소문자 구분 : aA,Aa,aa,AA 전부 다르게 인식한다.
- 길이 제한이 없다 : 백글자든 천글자든 설정할수는 있지만,

나중에 알아보기 힘들다!

(2) 예약어를 사용하면 안된다.

- 예약어 : 자바안에서 사용하는 다른 명령어

ex) 주요 예약어

abstract	default	if	package	this
assert	do	goto	private	throw
boolean	double	implements	protected	throws
break	else	import	public	transient
byte	enum	instanceof	return	true
case	extends	int	short	try
catch	false	interface	static	void
char	final	long	strictfp	volatile
class	finally	native	super	while
const	float	new	switch	
continue	for	null	synchronized	

- 명령어를 쓰는 키워드를 변수로 쓰면 안된다.

(3) 숫자로 시작하면 안된다.

(4) 특수문자는 '\_'와 '\$'만을 허용한다.

(5) 여러 단어 이름은 단어의 첫 글자를 대문자로 한다

EX)ageOfVampire,userName : 카멜표기법, 낙타봉.

## 6. 변수 초기화

- 변수를 사용하기 전에 처음으로 값을 저장하는 것

→ 설명) 변수의 선언을 통해 메모리에 공간을 할당하고,

그 공간 안에 값을 처음으로 집어넣는 것을 변수의 초기화라고 지칭함

→ 지역변수는 반드시 초기화 해야 된다.

✓ 선언 후 초기화

```
int age;  
age = 100;
```

✓ 선언과 동시에 초기화

```
int age = 100;
```

1) 선언후 초기화

int age; → 변수선언해서 방을 만들고, 방이름은 age, 타입은 int(정수)

→ 4byte

age =100; → 값을 100으로 넣음

2)선언과 동시에 초기화

```
int age = 100;
```

'=의 의미'

- 일반적 수학 = 은 등호고 '같다'는 의미,
- 컴퓨터에서의 '='은 대입시키겠다는 의미

ex) a=b라고 했을 때, a에 b의 값을 집어 넣겠다!

- 값 대입과 리터럴

## ✓ 값 대입

생성한 변수(저장 공간)에 값을 대입하는 것

```
int age;  
age = 10;  
age = 20;
```

\* 변수는 한 개의 데이터만 보관, 마지막에 대입한 값만 보관

## ✓ 리터럴

변수에 대입되는 값 자체를 의미한다.

```
short s = 32767;  
int i = 100;  
long l = 10000L;  
float f = 0.123f;  
double d = 3.14;
```

```
char c = 'A';  
String str = "ABC";
```

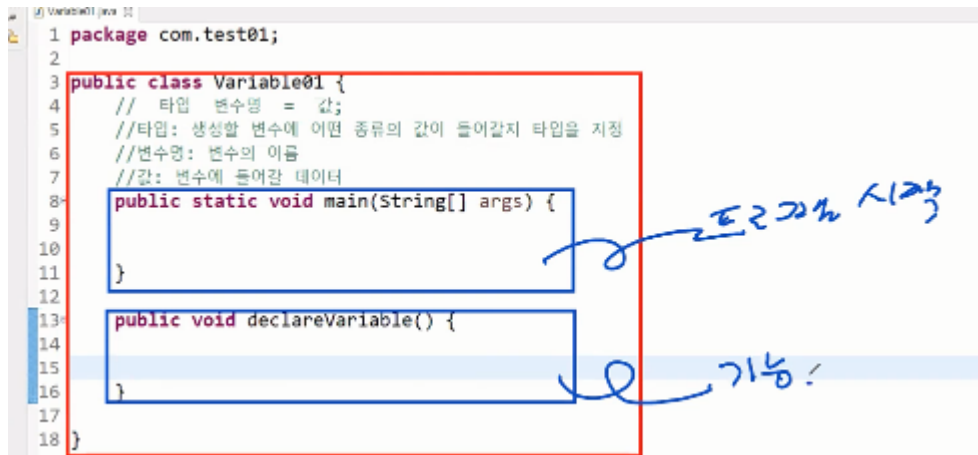
//4교시

Hello firstProject 출력하기

```
package firstProject.com.test;  
  
public class Hello {  
    / *main method : 프로그램의 진입점*/  
    public static void main(String[] args) {  
        / *실행 : ctrl + f11*/  
  
        System.out.println("Hello firstProject");  
    }  
}
```

//5교시





•

```
package com.test01;

public class Variable01 {
    //타입 변수명 = 값;
    //타입 : 생성할 변수에 어떤 종류의 값이 들어갈지 타입을 지정하는 것을 의미한다.
    //변수명 : 변수(방)의 이름, 내가 알아서 이해할수 있을 정도면 ok
    //값(=리터럴): 변수에 들어갈 데이터

    public static void main(String[] args) {

        //1. 메소드 실행 준비
        Variable01 test = new Variable01();
        //준비과정은 어떻게 해?
        //-->클래스명 변수명 = new 클래스명();

        //2. 메소드 실행
        test.declareVariable();
        //변수명. 메소드명();

    }

    public void declareVariable() {
        //숫자
        //정수

        byte bnum;
        short snum;
        int inum;
        long lnum;

        //실수(숫자)
        float fnum;
        double dnum;

        //논리형
        boolean isTrue;

        //문자형(a,b,c 각 하나씩을 의미)
```

```

char ch;

//문자열(참조형)
String str;

//-----
bnum = 1;
snum = 2;
inum = 4;
lnum = 8L;

fnum = 4.0f;
dnum = 8.0;

isTrue = true; //true or false

ch = 'A';
str = "A";

//-----
System.out.println(bnum);
System.out.println(snum);
System.out.println(inum);
System.out.println(lnum);

System.out.println(fnum);
System.out.println(dnum);

System.out.println(isTrue);

System.out.println(ch);
System.out.println(str);

}

}

```

//6교시