

제2장 변수(Variable)



1. 변수란(Variable)?

- 변하는 수?
- **하나의 값을 저장할 수 있는 메모리의 기억공간**
- 새로운 값이 저장되어지면? 어떻게 될까?

* 변수는 오로지 하나의 값만 저장하는 기능을 한다.

2. 변수의 타입(Data Type)

값(Data)

가, 나, 다, A, B, C..

문자 - char

숫자

- ① 정수(1, 2, 3 ...) – byte, short, int, long
- ② 실수(1.1, 3.14...) – float, double

논리 – boolean(**true**, **false**)

* 위에 8개의 타입을 기본형 변수(Primitive Variable)라 한다.

2. 변수의 타입(Data Type)

1. 기본형 변수(Primitive type Variable)

- 8개 (boolean, char, byte, short, int, long, float, double)

- 실제 값을 저장

• 아스키코드 표 참고.

• 'A' = 65, 'a' = 97

f : 16진수(10진수 15), 2진수 : 1111

16진수 1자리는 4bit가 필요함.

2. 참조형 변수(Reference type Variable)

- 기본형을 제외한 나머지(String, System 등)

- 객체의 주소를 저장(4 byte, 0x00000000~0xffffffff)

- C의 point개념

f : 4bit, 주소를 나타낼 때 16진수로 표기하는 것이 관례
f * 8 = 32bit(4byte)

2. 변수의 타입(Data Type)

기본형(Primitive type)

- 논리형 - true와 false중 하나를 값으로 갖으며, 조건식과 논리적 계산에 사용.
- 문자형 - 문자를 저장하는데 사용되며, 변수 당 하나의 문자만을 저장할 수 있음.
- 정수형 - 정수 값을 저장하는데 사용된다. 주로 사용하는 것은 int와 long이며, byte는 이진데이터를 다루는데 사용되며, short은 c언어와의 호환을 위해 추가됨.
- 실수형 - 실수 값을 저장하는데 사용된다. float와 double이 있음.

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

※ 이 표는 무조건 암기

2. 변수의 타입(Data Type)

1 bit

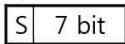


8 bit = 1 byte



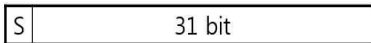
0,1을 저장하는 최소단위 2의 8승 = 256

byte $-2^7 \sim 2^7-1$



좌측 1bit는 부호bit
(0 : 양수, 1: 음수)

int $-2^{31} \sim 2^{31}-1$



자바에서는 메모리관리가 필요 없으므로 정수를
저장할 때 int형을 써주는 것이 좋다.

float $1+8+23=32$ bit = 4 byte

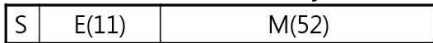


지수

가수

ex) $42.185 = 0.42185(\text{가수}) * 10^2 (\text{지수})$

double $1+11+52=64$ bit = 8 byte



3. 변수의 선언 방법

데이터타입 변수명;

```
int number;
```

```
number = 100;
```

```
int number = 100; (선언과 동시에 초기화)
```

```
String str = new String("boy"); (C는 문자열이 없다)
```

```
str = null;
```

4. 명명 규칙(Naming Convention)

1. 대소문자가 구분되며, 길이에 제한이 없다.
 - True와 true는 서로 다른 것으로 간주된다.
2. 예약어(Reserved word)를 사용해서는 안 된다.
 - true는 예약어라 사용할 수 없지만, True는 가능하다.
3. 숫자로 시작해서는 안 된다.
 - top10은 허용하지만, 7up은 허용되지 않는다.
4. 특수문자는 ‘ _ ’ 와 ‘ \$ ’ 만을 허용한다.
 - \$sharp은 허용되지만 S#arp는 허용되지 않는다.

4. 명명 규칙 – 권장 사항

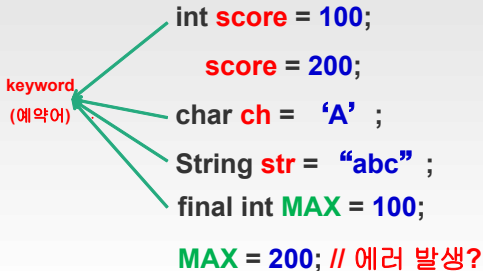
1. 클래스 이름의 첫 글자는 항상 대문자로 한다.
 - 변수와 메서드 이름의 첫 글자는 항상 소문자로 한다.
2. 여러 단어 이름은 단어의 첫 글자를 대문자로 한다.
 - LastIndexOf, StringBuffer, StringBuilder 등
3. 상수의 이름은 대문자로 한다. 단어는 ‘_’ 로 구분한다.
 - PI, MAX_NUMBER

5. 변수, 상수, 리터럴

- **변수(variable)** – 하나의 값을 저장하기 위한 공간
- **상수(constant)** – 한 번만 값을 저장할 수 있는 공간
- **리터럴(literal)** – 그 자체로 값을 의미하는 것

keyword
(예약어)

```
int score = 100;  
score = 200;  
char ch = 'A' ;  
String str = "abc" ;  
final int MAX = 100;  
MAX = 200; // 에러 발생?
```

A diagram with a central point labeled 'keyword (예약어)' in red. Four green lines radiate from this point to the left side of four lines of code: 'int', 'score', 'char', and 'String'. These words are highlighted in red in the original image to show they are keywords.

6. 리터럴과 접미사

int bin = 0b10; 2진수

int oct = 0100; 8진수

int hex = 0x100; 16진수

long l = 1000000000000L;

float f = 3.14f

double d = 3.14d

float f = 100f;

10. → 10.0 double형

.10 → 0.10 (디폴트)

10f → 10.0f

3.14e3f → 3140.0f

* 리터럴은 그 자체로서의 값을 의미하고, L,f,d 등은 접미사의 역할을 함.

7. 변수의 기본값과 초기화

♠ 변수의 초기화 : 변수에 처음으로 값을 저장하는 것

- 지역변수(메서드 내 선언되어 있는 변수)는 사용되기 전에 반드시 초기화 해주어야 한다.

자료형(Data type)	기본값(Default Value)
boolean	false
char	'\u0000' -> 유니코드
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0
참조형 변수	null

▷ 좌측 표는 class내에 선언되어지는 멤버변수들에 한해서 초기화가 된다는 것이다.

▷ 기본적으로 변수는 초기화가 원칙이다.

8. 문자와 문자열

```
char ch = 'A';
```

```
char ch = 'AB'; // 에러
```

```
String s1 = "AB";
```

```
char ch = ""; // 에러
```

```
String s1 = "";
```

```
String s1 = "A" + "B"; // "AB"
```

`"" + 7` → `"" + "7"` → `"7"`
숫자를 문자로 변환

`"" + 7` → 문자열 7이 됨.

`7 + 7 + ""` → 문자열 14가 됨

문자열 + any type

문자열

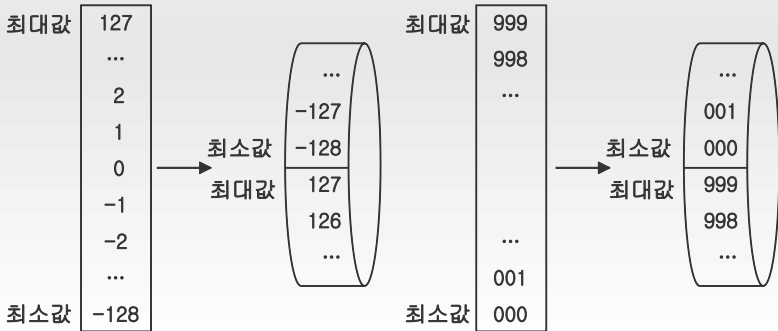
any type + 문자열

문자열

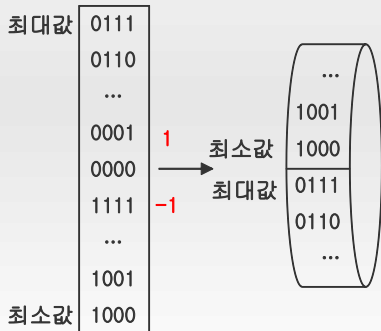
9. 정수의 오버플로우(Overflow) 현상

byte b = 127; byte b = 128; //에러

b = (byte)(b + 1); // b에 저장된 값을 1증가(값이 -128이 된다)



10. 2의 보수값 – 우리가 생각하는 마이너스 값



보수란 ? 0을 1로, 1을 0으로~
1. 1의 보수를 취함.
2. +1을 해줌

11. 형 변환(Casting)

형변환이란?

- 해당하는 데이터 타입의 값을 다른 타입으로 변환하는 것을 칭함.
- boolean을 제외한 7개의 기본형은 서로 형변환이 가능함.

```
float f = 1.6f;
```

```
int i = (int)f;
```

변 환	수 식	결 과
int → char	(char)65	'A'
char → int	(int)'A'	65
float → int	(int)1.6f	1
int → float	(float)10	10.0f

1. byte → int 자동형 변환

```
byte b = 10;
```

```
int i = (int)b; // 생략가능
```

2. int → byte 강제형 변환

```
int i = 300;
```

```
byte b = (byte)i; // 생략불가
```


감사합니다.

