

변수와 자료형

수업목표

1. 변수의 선언, 명명규칙과 초기화의 의미를 설명할 수 있다.
2. 변수와 컴퓨터 메모리의 구조를 연관지어 설명할 수 있다.

$$3x + 2 = 8$$

변수

variable

Data와 정보

데이터(Data)

- 데이터란 관찰의 결과로 나타난 정량적 혹은 정성적인 실제 값을 의미한다.

정보(Information)

- 정보란 데이터를 기반으로 하여 의미를 부여한 것이다.

Data와 정보

- 예1) 에베레스트의 높이 : 8848m (Data)
에베레스트는 세계에서 가장 높은 산이다. (정보)
- 예2) 월간 아이스크림 판매량 (DATA)
아이스크림은 겨울보다 여름에 잘 팔린다. (정보)
- 예3) 2018년 8월 1일 서울의 온도 : 39.6°C (DATA)
2018년 여름은 111년 사상 최악의 폭염이다. (정보)

kumgangMt

1638

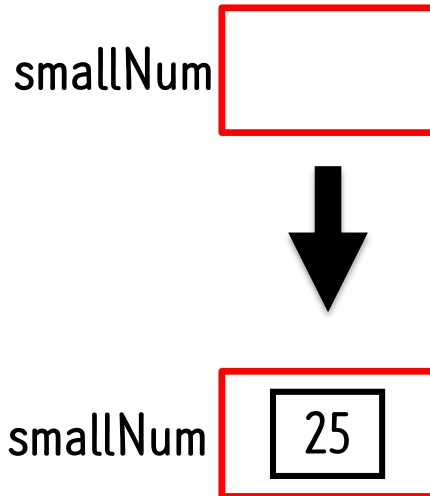
ChimborazoMt

6268

everestMt

8848

변수의 선언



```
int smallNum ;
```

↓ ↘

데이터타입 지정 변수명 지정

변수의 초기화

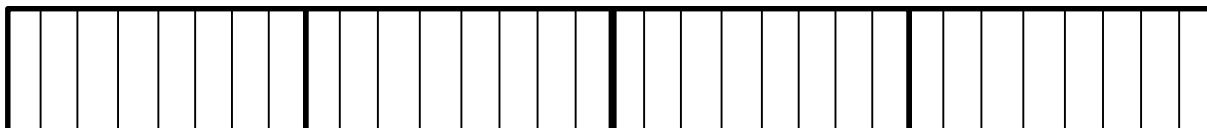
```
int smallNum = 25;
```

* smallNum 변수를 25로 초기화

Memory(RAM)



```
int smallNum;
```



변수의 선언 / 변수의 초기화

- 메모리 상에 데이터를 저장할 수 있는 공간을 할당하는 것

```
int smallNum ;
```

- 변수를 사용하기 전에 처음으로 값을 저장하는 것

```
int smallNum = 25;
```


실습해보기

```
public static void main(String[] args) {  
    int kh;  
    kh = 1;  
    System.out.println(kh+1);  
  
    kh = 2;  
    System.out.println(kh+1); // kh는 변수, 변하는 수  
}
```

필수사항

1. 대소문자가 구분되며 길이 제한이 없다.
2. 예약어를 사용하면 안 된다.
3. 숫자로 시작하면 안 된다.
예) age1은 가능하지만 1age는 불가능
4. 특수문자는 ‘_’와 ‘\$’만을 허용한다.
예) sh@rp는 불가능하지만 \$harp는 가능

권장사항 - 가독성

1. 클래스 이름의 첫글자는 항상 대문자!
- 변수와 메서드 첫글자는 항상 소문자!
2. 여러 단어 이름은 단어의 첫 글자를 대문자
예) lastIndexOf, stringBuffer
3. 상수의 이름은 대문자, 단어는 _로 구분한다
예) MAX_VALUE

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	

값의 종류와 표현방법

종류	의미
정수	소수점을 포함하지 않은 수
실수	소수점을 포함한 수
문자	' ' 안에 표현한 문자
문자열	" " 안에 표현한 문자

값의 종류와 표현방법

정수값		실수값	
26	0x1A	112.4	112.4f
032	0b11010	112.4d	1.124e2
문자값		진리값	
'A'	'\u0065'	true	false

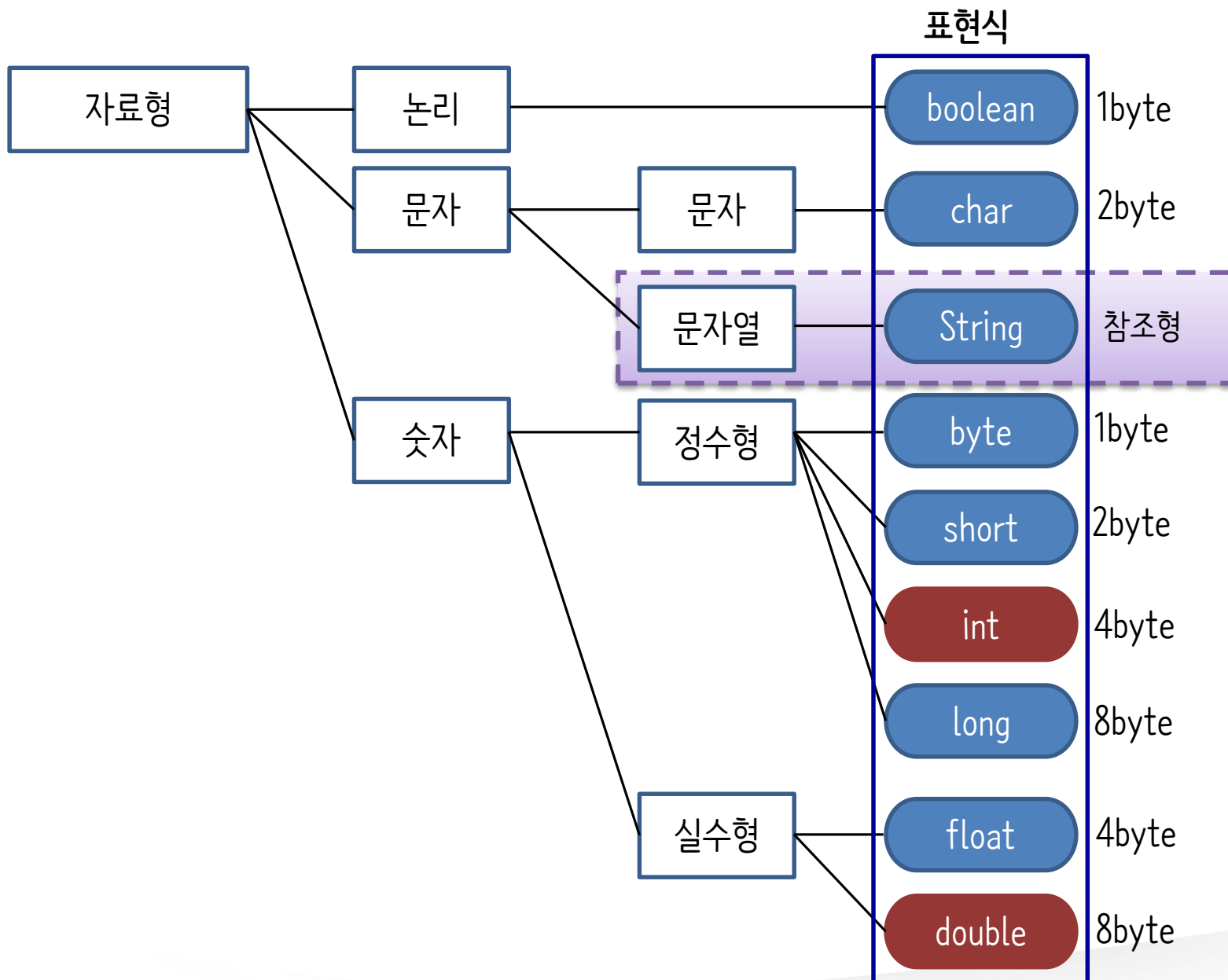
- ※ 문자는 1글자만 표기할 수 있고, 문자열은 1글자 이상을 표기할 수 있음
- ※ 문자는 사실상 하나의 숫자값임 (아스키코드값 기억하세요!)

변수의 자료형

ASCII 코드

10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자	10진수	16진수	문자
0	0X00	NULL	32	0x20	SP	64	0x40	@	96	0x60	`
1	0X01	SOH	33	0x21	!	65	0x41	A	97	0x61	a
2	0X02	STX	34	0x22	"	66	0x42	B	98	0x62	b
3	0X03	ETX	35	0x23	#	67	0x43	C	99	0x63	c
4	0X04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d
5	0X05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e
6	0X06	ACK	38	0x26	&	70	0x46	F	102	0x66	f
7	0X07	BEL	39	0x27	'	71	0x47	G	103	0x67	g
8	0X08	BS	40	0x28	(72	0x48	H	104	0x68	h
9	0X09	HT	41	0x29)	73	0x49	I	105	0x69	i
10	0X0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j
11	0X0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k
12	0X0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l
13	0X0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m
14	0X0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n
15	0X0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o
16	0X10	DLE	48	0x30	0	80	0x50	P	112	0x70	p
17	0X11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q
18	0X12	SC2	50	0x32	2	82	0x52	R	114	0x72	r
19	0X13	SC3	51	0x33	3	83	0x53	S	115	0x73	s
20	0X14	SC4	52	0x34	4	84	0x54	T	116	0x74	t

변수의 자료형



변수의 자료형(Type)

기본형(Primitive type)

- 실제 데이터(값)를 저장
- 논리형, 문자형, 정수형, 실수형으로 나뉘지고, 8개의 자료형이 있다.
- 각 자료형별 데이터 저장크기가 다름

참조형(Reference type)

- 데이터가 저장되어 있는 주소를 저장(객체의 주소)
- 기본형을 제외한 나머지(String 등), 사용자정의 자료형
- 4byte의 공간을 저장공간으로 할당

저장공간이 제한적이기 때문에 데이터를 저장크기에 대한 기준과 CPU가 데이터를 처리할때 일정한 기준이 필요하기 때문에 저장단위를 구성

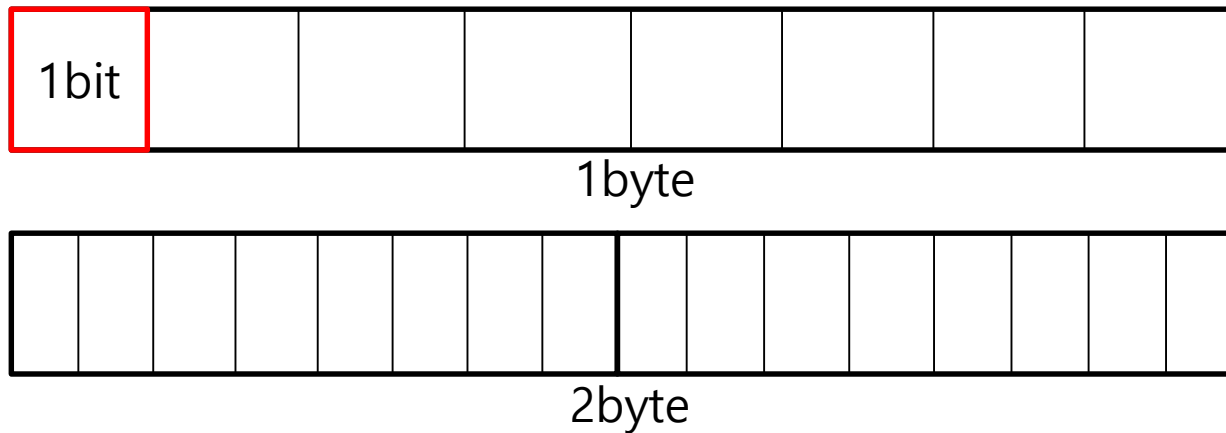
비트(bit)

컴퓨터가 나타내는 **데이터의 저장** 최소단위로서 **2진수 값 하나**를 저장할 수 있는 메모리공간을 의미

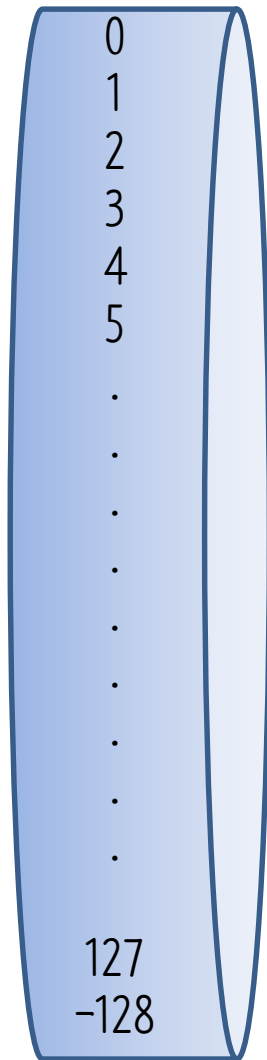
데이터 저장단위

바이트(byte)

데이터 처리 또는 문자의 **최소 단위**로서 8개의 비트가 모여 하나의 바이트가 구성됨

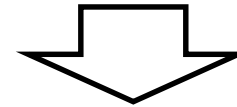


데이터 오버플로우



0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

127+1을 하면 범위를 초과한 128이 되고 허용된 범위이상의 비트를 침범하게 되는데 이를 오버플로우라고 한다.



byte형

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

byte형 허용범위 최소값인 -128이 되는 것이다.

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

$$-128 \sim 127$$

자료형 small int ;

자료형	범 위	크기(byte)
boolean	true, false	1
char	0~65,535(유니코드문자)	2
byte	-128 ~ 127	1
short	-32,768 ~ 32,767	2
int	-2,147,483,648 ~ 2,147,483,647	4
long	-9,223,372,036,854,775,808 ~ 9,223,372,036,854,775,807	8
float	$\pm 1.4E-45 \sim 3.4E38$	4
double	$\pm 4.9E-324 \sim 1.8E308$	8

컴퓨터는 2진수로 인지하기 때문에 2^n (n=비트크기)으로 범위가 할당됨.

기본형

분 류	종 류	크 기	표현범위	내 용
논리형	boolean	1byte	true/false	true,false중 하나의 값을 가짐 조건식이나 논리적 계산에 사용
문자형	char	2byte	0 ~ 65,535	한 개의 문자를 저장하는데 사용 Unicode인코딩 방식사용(2byte)
정수형	byte	1byte	-128 ~ 127	이진데이터를 다루는데 사용
	short	2byte	-32768 ~ 32767	C언어와 호환용.
	int	4byte	-2147483648 ~ 2147483647	정수를 저장하는데 사용
	long	8byte	...	☞ 주로 사용 / 기본 int형
실수형	float	4byte	1.4E-45 ~ 3.4E038	실수를 저장하는데 사용
	double	8byte	4.9E-324 ~ 1.8E308	실수를 표현하는데 한계 -> 오차발생 ☞ 기본 double형

변수의 선언(예시)

자료형

변수명

;

```
//논리형 변수 선언  
boolean bool;
```

```
//문자형 변수 선언  
char ch;
```

```
//문자열 변수 선언  
String str;
```

```
//정수형 변수 선언  
byte bnum;  
short snum;  
int inum;  
long lnum;
```

```
//실수형 변수 선언  
float fnum;  
double dnum;
```

대표적인 데이터의 종류

```
System.out.println(1); // 숫자 1  
System.out.println(1.1); // 실수 1.1  
System.out.println('1'); // 문자 1  
System.out.println("1"); // 문자열 1
```


변수의 자료형 - 숫자와 문자

이해를 돕기 위한 예제 - 무엇이 출력 될까?

```
System.out.println(1 + 1);  
System.out.println(1.1 + 1.1);  
System.out.println('1'+'1');  
System.out.println("1" + "1");
```

이해를 돕기 위한 QUIZ - 무엇이 출력 될까?

```
System.out.println("Hello" + 'q');  
System.out.println("Hello" + 3.14);  
System.out.println(10+20+"Hello");  
System.out.println(10+"Hello"+20);
```

```
System.out.println("Hello" + 10);  
System.out.println(10+"Hello");  
System.out.println(10+(20+"Hello"));
```

실습 예제1 - 변수

◆ 각 자료형별 변수를 만들고 초기화한 후 그 값을 출력하는 코드를 작성해보자.

- 정수형 100, 9999억,
- 실수 486.520(float), 567.890123
- 문자 A
- 문자열 Hello world,
- 논리 true

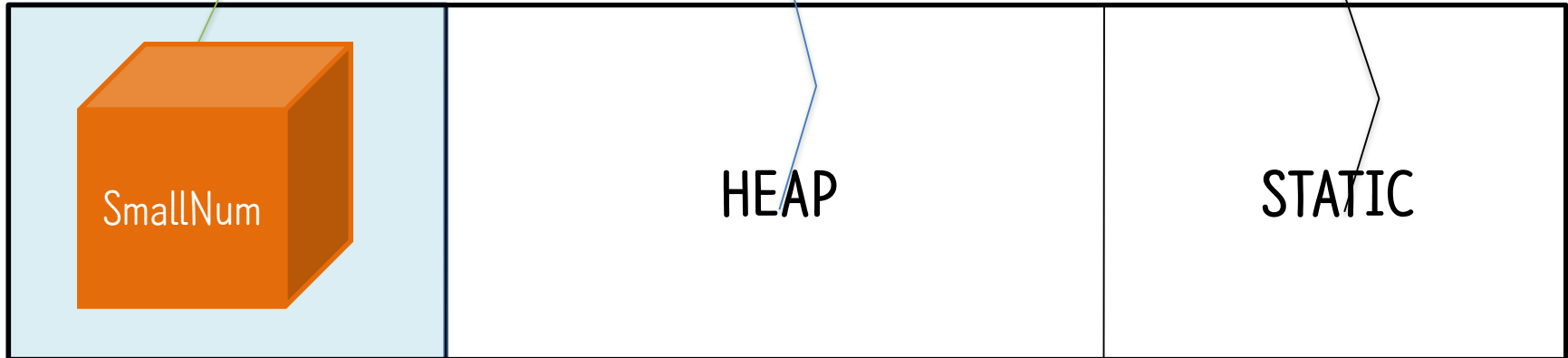
단, 변수를 이용하여 출력하시오 .

변수와 메모리구조

메소드를 호출하면 자동으로
생기고 메소드가 끝나면 자동소멸
(지역변수, 매개변수, 메소드 호출
스택 등)

new연산자에 의해 동적으로 할당하고
저장되는 공간, 객체, 배열 등

static예약어로 선정된 필드,
메소드가 저장되는 공간
클래스 변수 등



RAM 구조

상수란?

수학에서 변하지 않는 값을 의미하고, 컴퓨터에서는 한번만 저장할 수 있는 공간으로 초기화 이후 다른 데이터(값)을 대입할 수 없다.

상수선언

```
final int AGE = 20;
```

↓
상수선언
예약어

↓
상수선언
명명관례

리터럴이란?

변수에 대입되는 값 자체를 의미

short s = 32767;

int i = 100;

long l = 100000000L;

float f = 0.123f;

double d = 3.14;

char c = 'A';

String str = "ABC";