

변수(variable)



변수를 왜 써요?

클라이언트 요구사항 :

구구단을 홈페이지에 출력해주세요.

화면출력 :

== 2단 ==

$$2 * 1 = 2$$

$$2 * 2 = 4$$

$$2 * 3 = 6$$

$$2 * 4 = 8$$

$$2 * 5 = 10$$

$$2 * 6 = 12$$

$$2 * 7 = 14$$

$$2 * 8 = 16$$

$$2 * 9 = 18$$

== 3단 ==

$$3 * 1 = 3$$

$$3 * 2 = 6$$

$$3 * 3 = 9$$

$$3 * 4 = 12$$

$$3 * 5 = 15$$

$$3 * 6 = 18$$

$$3 * 7 = 21$$

$$3 * 8 = 24$$

$$3 * 9 = 27$$

.

.

.

.

.

.

.

.

== 9단 ==

$$9 * 1 = 9$$

$$9 * 2 = 18$$

$$9 * 3 = 27$$

$$9 * 4 = 36$$

$$9 * 5 = 45$$

$$9 * 6 = 54$$

$$9 * 7 = 63$$

$$9 * 8 = 72$$

$$9 * 9 = 81$$



결과는 똑같은데...

개발자A :

```
System.out.println("==2단==");
System.out.println("2 * 1 = 2");
System.out.println("2 * 2 = 4");
System.out.println("2 * 3 = 6");
System.out.println("2 * 4 = 8");
System.out.println("2 * 5 = 10");
System.out.println("2 * 6 = 12");
System.out.println("2 * 7 = 14");
System.out.println("2 * 8 = 16");
System.out.println("2 * 9 = 18");
```

```
System.out.println("==3단==");
System.out.println("3 * 1 = 3");
System.out.println("3 * 2 = 6");
System.out.println("3 * 3 = 9");
System.out.println("3 * 4 = 12");
System.out.println("3 * 5 = 15");
System.out.println("3 * 6 = 18");
System.out.println("3 * 7 = 21");
System.out.println("3 * 8 = 24");
System.out.println("3 * 9 = 27");
```

```
System.out.println("==4단==");
System.out.println("4 * 1 = 4");
System.out.println("4 * 2 = 8");
System.out.println("4 * 3 = 12");
System.out.println("4 * 4 = 16");
System.out.println("4 * 5 = 20");
System.out.println("4 * 6 = 24");
System.out.println("4 * 7 = 28");
System.out.println("4 * 8 = 32");
System.out.println("4 * 9 = 36");
```

:

```
System.out.println("==9단==");
System.out.println("9 * 1 = 9");
System.out.println("9 * 2 = 18");
System.out.println("9 * 3 = 27");
```

결과는 똑같은데...

개발자B :


```
for(int dan = 2; dan<10; dan++){  
    System.out.println("==" + dan + " 단 ==");  
    System.out.println(dan + " * 1 = " + (dan * 1));  
    System.out.println(dan + " * 2 = " + (dan * 2));  
    System.out.println(dan + " * 3 = " + (dan * 3));  
    System.out.println(dan + " * 4 = " + (dan * 4));  
    System.out.println(dan + " * 5 = " + (dan * 5));  
    System.out.println(dan + " * 6 = " + (dan * 6));  
    System.out.println(dan + " * 7 = " + (dan * 7));  
    System.out.println(dan + " * 8 = " + (dan * 8));  
    System.out.println(dan + " * 9 = " + (dan * 9));  
}
```



결과는 똑같은데...

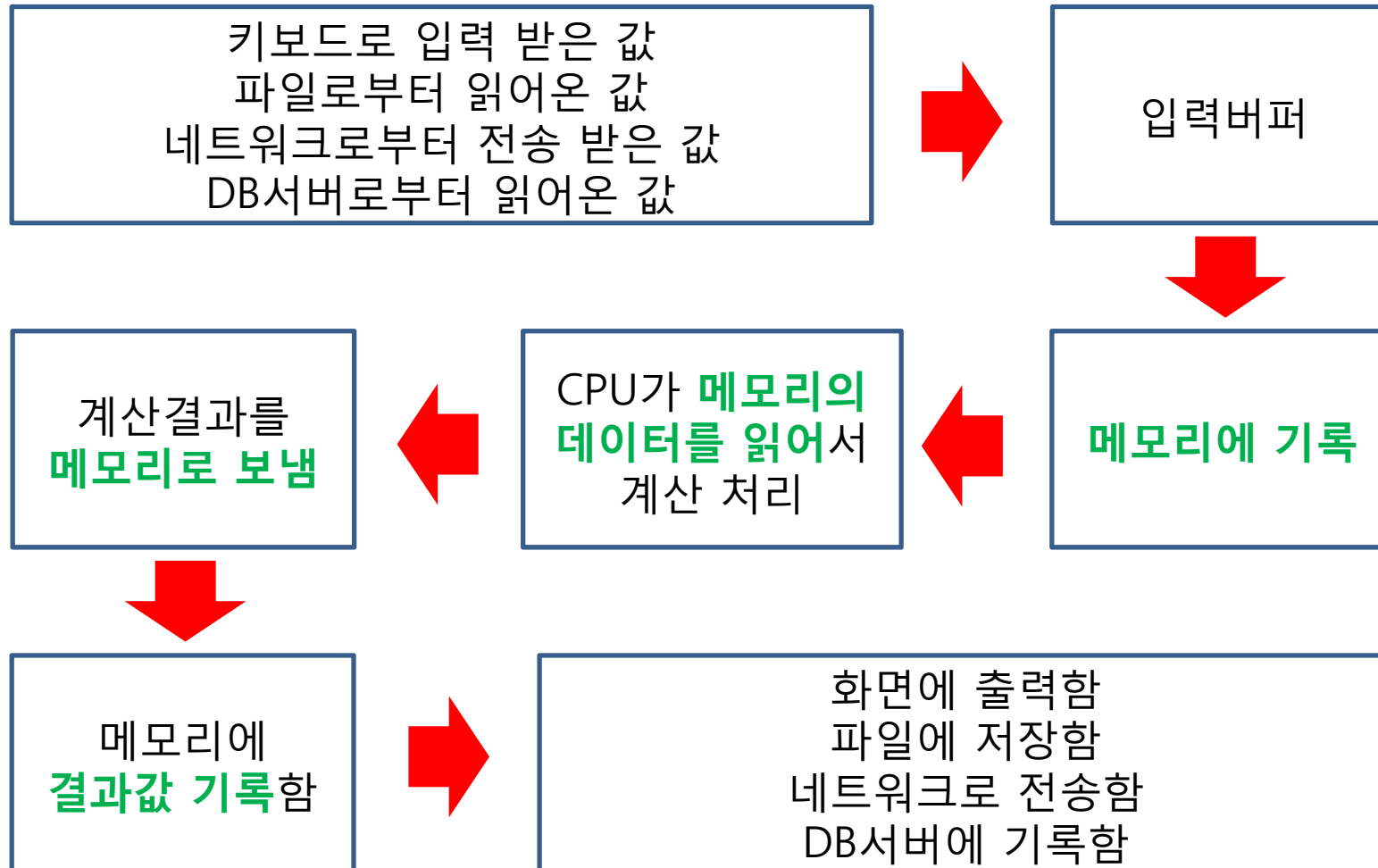
개발자C :

```
for(int dan = 2; dan<10; dan++){  
    System.out.println("==" + dan + " 단==");  
    for(int i = 1; i < 10; i++){  
        System.out.println(dan + "*" + i + " = " + (dan*i));  
    }  
}
```





프로그램 작동 원리





정수 양수, 음수 표현

정수표현(양수)

부호비트

+, - 표현

0일때 양수

1일때 음수

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

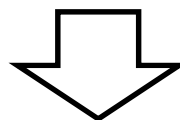
1

정수의 데이터표현

정수표현(음수)

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

1과 더하기 연산시 0에 값이 나오지 않음.



-1

1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

정수의 2의 보수를 이용하여 음수표현





실수 데이터 표현방식

실수표현

IEEE754 표준방식에 따라 표현

float 32bit = 1+8+23 (4byte) 정밀도 7자리

S	E(8)	M(23)
---	------	-------

double 64bit = 1+11+52 (8byte) 정밀도 15자리

S	E(11)	M(52)
---	-------	-------

2진수로 실수를 모두 표현하기에 한계가 있어 **근사값을 구해서 표현**하는
것으로 정의

오차범위를 줄이려면 정확도가 높은 double형으로 작성






변수

변수란?

메모리 공간(RAM)에 **한 개의 값을 기록**하기 위한 장소(공간)



JAVA에서
변수는 반드시
선언 후에,
값을 대입해야한다.





변수의 선언

변수의 선언

메모리 상에 데이터를 저장할 수 있는 공간을 할당하는 것

자료형



데이터타입지정

변수명



변수명 지정

; 마침

```
int num;
```





변수의 자료형(Type)

기본형(Primitive type)

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

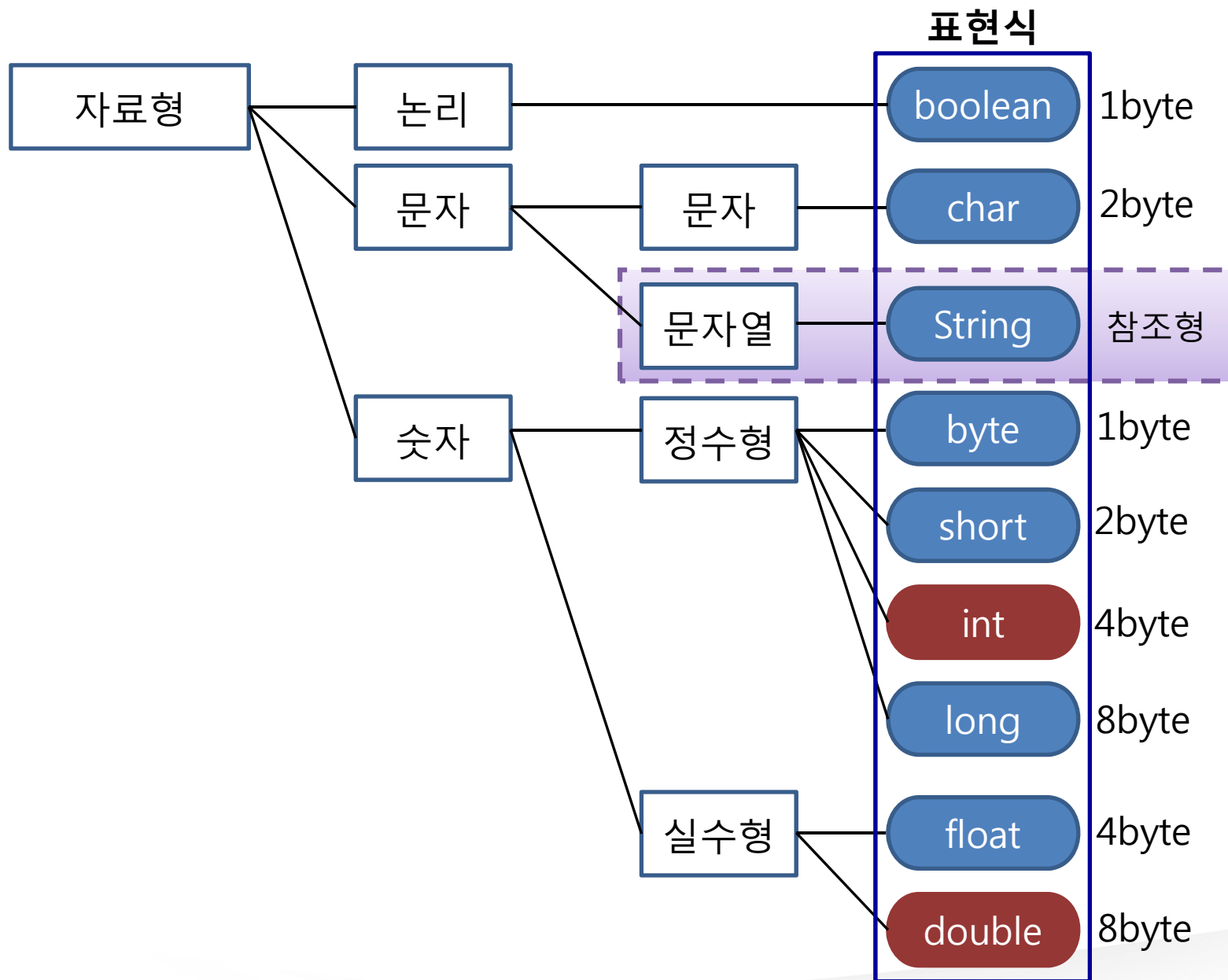
참조형(Reference type)

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





기본 자료형





데이터 저장단위

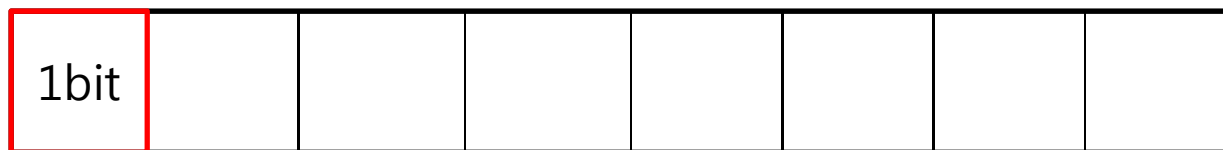
저장공간이 제한적이기 때문에 데이터를 저장크기에 대한 기준과 CPU가 데이터를 처리할때 일정한 기준이 필요하기 때문에 저장단위를 구성

비트(bit)

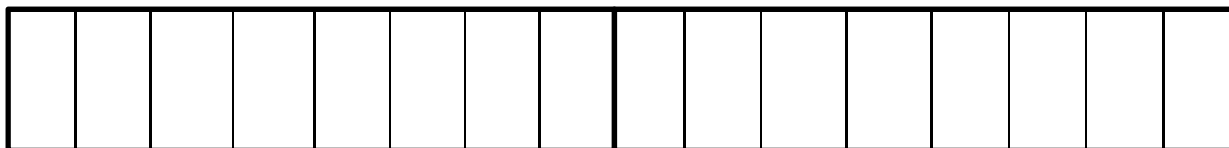
컴퓨터가 나타내는 **데이터의 저장** 최소단위로서 **2진수 값 하나**를 저장할 수 있는 메모리공간을 의미

바이트(byte)

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



1byte



2byte



저장가능범위

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	$\pm 1.4\text{E}-45 \sim 3.4\text{E}38$	4
double	$\pm 4.9\text{E}-324 \sim 1.8\text{E}308$	8

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





자료형 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

데이터타입지정

변수명

변수명 지정

; 마침

//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;





변수의 명명규칙

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 \$sharp는 가능
5. 여러 단어 이름은 단어의 첫 글자를 대문자로 한다.
예) ageOfVampire, userName 낙타표기법(Camel Case)
6. 한글입력 가능하다.
7. 의미있는 변수명을 사용하길 추천한다.





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	



변수에 값대입

예) int age;
age = 10;

등호가 아닌 대입연산자

리터럴 Literal

변수에 대입되는 값 자체를 의미

short s = 32767;

int i = 100;

long l = 100000000L;

float f = 0.123f;

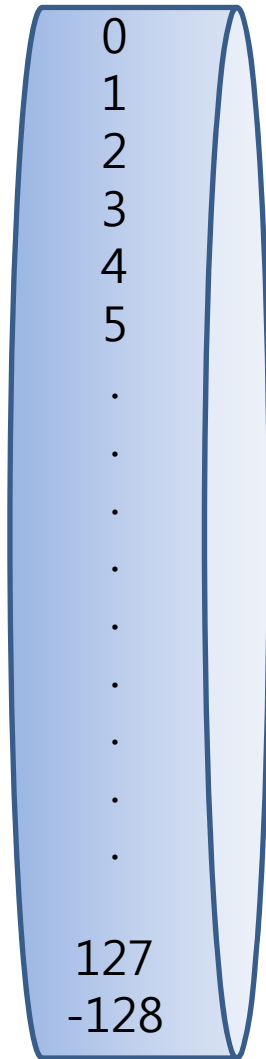
double d = 3.14;

char c = 'A';

String str = "ABC";



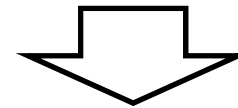
데이터 오버플로우



byte형

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

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



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

byte형 허용범위 최소값인 -128이 되는 것이다.





변수의 초기화

변수의 초기화

변수를 선언하는 시점에 값을 대입하는 것.

☞ 지역변수는 반드시 값대입후에 사용해야한다.

선언과 동시에 초기화

```
int age = 100 ;
```





상수선언



문자열 표현

컴퓨터에서 “기차”, “출력하세요” 등과 같이 단어나 문장을 문자열이라고 표현하고, “”로 묶여 있으면 문자열로 인식한다.
Java에서는 String객체를 이용하여 저장한다.

문자열 초기화

```
String ch = “기차”;  
String ch = new String(“기차”);  
String ch = “기차”+”칙칙폭폭”;  
String ch = new String(“기차”+”칙칙폭폭”);  
String ch = “기차”+123+45+”출발”;  
String ch = 123+45+ “기차”+”출발”;  
다른형 +”문자열 ” -> 문자열 / 문자열+ 다른형 -> 문자열
```





변수 사용 목적

변수를 사용하지 않으면

```
System.out.println(100 + 10);  
System.out.println((100 + 10) * 10);  
System.out.println(((100 + 10) * 10) - 10);  
System.out.println((((100 + 10) * 10) - 10) / 10);
```

변수를 사용하면

```
Int a = 100;  
System.out.println(a + 10);  
System.out.println((a + 10) * 10);  
System.out.println(((a + 10) * 10) - 10);  
System.out.println((((a + 10) * 10) - 10) / 10);
```





컴퓨터의 동작원칙

1. 반드시 같은 데이터타입 간에 연산을 한다.
2. 반드시 동일한 데이터타입의 값을 동일한 데이터타입의 변수에 대입한다.
3. 결과값 또한 동일한 데이터타입의 값을 가진다.

위의 기본원칙이 지켜지지 않을 경우 프로그래밍언어가 암묵적으로, 혹은 사용자에게 의해 명시적으로 **형 변환**을 해야한다.

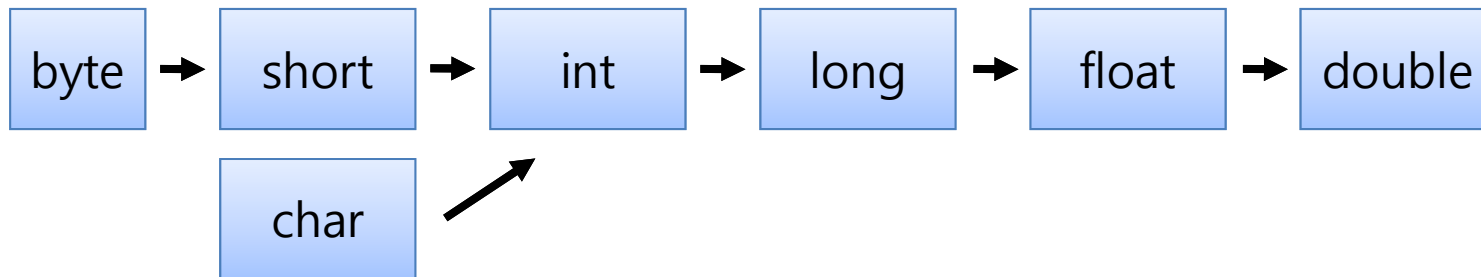




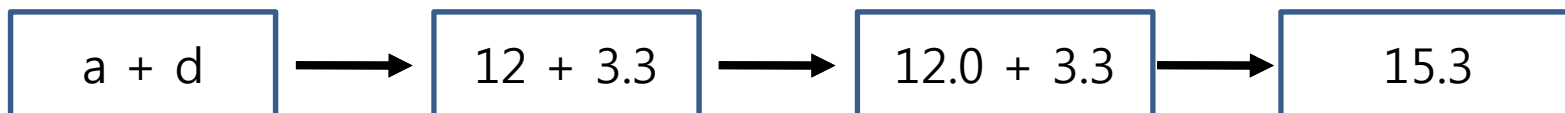
형변환(casting)

자동 형변환

연산시 컴파일러가 자동으로 형이 변환하는 것을 의미한다.



예시) `int a = 12;`
`double d = 3.3;`
`double result = a + d`





형변환(casting) 예제

자동형변환되는 경우

1. 자료형이 다른 값에 대입될 때.

```
byte b = 100;
```

```
int i = b;
```

2. 자료형이 다른 값이 계산될 때.

```
int a = 3;
```

```
double b = 5.7;
```

```
System.out.println(a+b); //자동으로 a가 double로 형변환 후 연산
```

강제형변환 해야하는 경우

```
int num = 2147483647;
```

```
int result = num+1;
```

```
System.out.println(result); //-2147483647
```

```
long result_ = (long)num + 1;
```

```
System.out.println(result_); //2147483648
```





형변환(casting)

byte와 short연산

자바에서는 byte와 short연산시 자동으로 결과값을 int형으로 반환한다.

```
byte bnum=10, bnum2=20;  
byte result= bnum+ bnum2; //에러!  
int result= bnum+ bnum2; //OK!!  
byte result= (byte)(bnum+ bnum2);  
//OK!!
```





형변환(casting)

강제 형변환

자료형으로 형변환을 해줄 수 있다. 데이터가 큰 자료형에서 작은 자료형으로 변경 시 데이터 손실이 있을 수 있어 유의해야 한다.

```
double temp;  
int age = (int)temp ;
```





데이터 손실

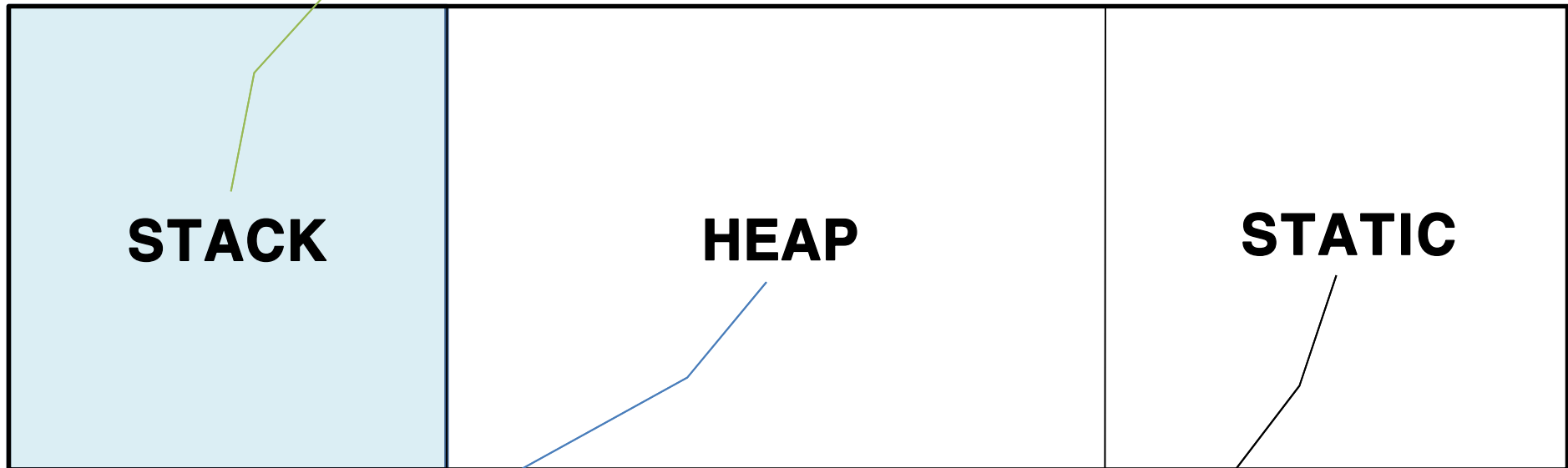




변수와 메모리구조

RAM 구조

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





System.out.print()

()안의 변수, 문자, 숫자, 논리값을 모니터에 출력해주는 메소드

예 : `System.out.print("안녕하세요");`, `System.out.print(123);`,
`System.out.print(변수명);`

System.out.println()

Print와 동일하게 출력해주지만 출력 후 자동으로 출력창에 줄바꿈을 해주는 메소드

예 : `System.out.println("안녕하세요");`, `System.out.println(123);`,
`System.out.println(변수명);`





출력메소드

System.out.printf("%형식", 변수 등)

정해져 있는 형식에 맞춰서 그 형식에 맞는 값(변수)을 줄바꿈 하지 않고 출력

정수표현

- %d : 정수형, %o : 8진수, %x : 16진수

실수표현

- %f : 실수(소수점 아래 6자리), %e : 지수형태표현
- %g : %e와 %f 중 더 짧은 표현을 사용
- %A : 16진수 실수

문자/문자열

- %c : 문자, %s : 문자열

불린형

- %b : 논리형

너비 및 정렬방법

- %5d : 5칸을 확보하고 오른쪽정렬
- %-5d : 5칸을 확보하고 왼쪽 정렬
- %.2f : 소수점 아래 2자리까지만 표시





escape문자

특수문자	문자 리터럴	비 고
tab	\t	정해진 공간만큼 띄어쓰기
new line	\n	출력하고 다음라인으로 옮김
역슬래쉬	\\	특수문자 사용시 백슬러시(\)를 넣고 특수문자를 넣어야 함
작은따옴표	'\''	
큰따옴표	'\"'	
유니코드	\u	유니코드 표시할때 사용





실습 예제 1

각 자료형별 변수를 만들고 초기화한 후 그 값을 출력하는 코드를 작성해 보자.

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

단, 변수를 이용하여 출력하시오 .





실습 예제 2

자신의 신상정보를 저장할 변수를 만들고 정보를 변수에 대입, 출력하는 프로그램을 작성하세요.

- 이름, 나이, 성별, 주소, 전화번호, 이메일

작성한 변수를 활용해 옆 친구의 신상정보를 다시 변수에 대입 출력하는 프로그램으로 수정하세요.

이름	나이	성별	지역	전화번호	이메일
홍두깨	22	남	서울	01012345678	myemail@email.com
고길동	50	남	경기도	01098765432	mrgogo@email.com





입력값 받기 위한 Scanner Class

Scanner Class : 사용자로부터 입력되는 정수, 실수, 문자열을 처리하는 클래스

import 작성

```
import java.util.Scanner;
```

Scanner생성

```
Scanner sc=new Scanner(System.in);
```

키보드 입력값 받기

1. 정수 : `sc.nextInt();`
2. 실수 : `sc.nextFloat();` 나 `sc.nextDouble();`
3. 문자열 : `sc.next();` 나 `sc.nextLine();`

`next()`는 띄어쓰기 입력불가
`nextLine()`은 문자열에 띄어쓰기 가능.





입력을 위한 Scanner Class

예제

```
1  import java.util.Scanner;
2
3  public class ScannerTest {
4      public static void main(String[] args){
5          Scanner sc = new Scanner(System.in);
6
7          System.out.print("int i =>");
8          int i = sc.nextInt();
9          System.out.print("float f =>");
10         float f = sc.nextFloat();
11         System.out.print("double d =>");
12         double d = sc.nextDouble();
13         System.out.print("char ch =>");
14         char ch = sc.next().charAt(0);
15         System.out.print("String str =>");
16         String str = sc.nextLine();
17     }
18 }
```





Scanner next(),nextLine() 오류해결하기

예제

```
1 public class ScannerTest_Error {  
2     public static void main(String[] args) {  
3         Scanner sc = new Scanner(System.in);  
4  
5         System.out.print("이름 => ");  
6         String userName = sc.next();  
7         System.out.print("성별=> ");  
8         char userGender = sc.next().charAt(0);  
9         System.out.print("주소 => ");  
10        sc.nextLine(); // ↵ 날리기용  
11        String userAddr = sc.nextLine();  
12  
13        System.out.printf("이름 : %s\n", userName);  
14        System.out.printf("성별 : %s\n", userGender);  
15        System.out.printf("주소 : %s\n", userAddr);  
16    }  
17  
18 }  
19
```





실습 예제 3

자신의 이름(String), 나이(int), 주소(String), 키(double), 몸무게(double), 성별(char)를 입력 받고 출력하는 프로그램을 작성하세요.

- 키는 정수로 변환하여 출력
- 몸무게는 첫째자리까지 출력





종합 실습예제1

1. 실행용 클래스

패키지 : kh.java.var

클래스명 : Run

추가메소드

```
public static void main(String[] args)
```

내용 : VarExample의 Example실행

기능제공 클래스

패키지 : kh.java.funtion

클래스명 : VarExample

추가 메소드

```
public void Example(){ }
```

내용 : 정수 두개를 입력받아 두수의 합, 차, 곱,
나누기 출력





종합 실습예제2

2. 1번 VarExample클래스에서 메소드 추가

추가 메소드

```
public void Example2(){ }
```

내용 : 가로, 세로 값을 실수형으로 입력 받아
사각형의 면적과 둘레를 계산하여 출력

계산공식

면적 : 가로 * 세로

둘레 : (가로 + 세로) * 2





종합 실습예제3

3. 1번 VarExample클래스에서 메소드 추가 추가 메소드

```
public void Example3(){ }
```

내용 : 영어 문자열 값을 키보드로 입력 받아
각 자리의 문자를 출력

출력 예) 입력 : apple
첫번째 문자 : a
두번째 문자 : p
세번째 문자 : p

