

#### JAVA 기본 API 클래스

java.lang 패키지는 자바 프로그램의 기본적인 클래스를 담고 있는 패키지이다.
( 따라서 java.lang 패키지에 있는 클래스등은 import 없이 사용할 수 있다. )

지금까지 사용한 String과 System 클래스도 java.lang 패키지에 포함되어 있어 import없이 사용했다.

java.lang 패키지에 속하는 주요 클래스에 대해서 알아보자.



# 기본 API 클래스

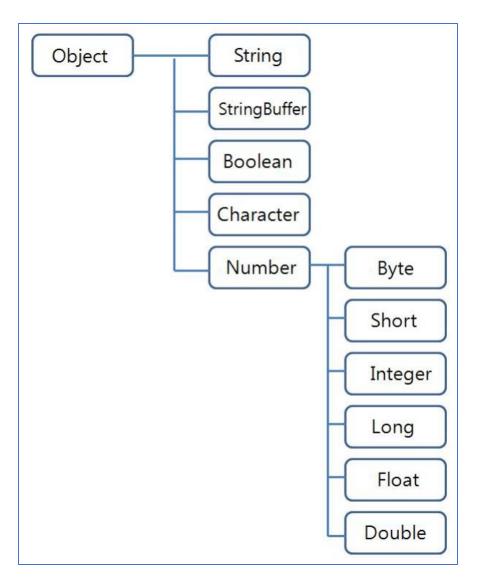
클래스		용도	
Object		자바 클래스의 최상위 클래스로 사용	
System		표준 입력장치(키보드)로부터 데이터를 입력받을 때 사용 표준 출력장치(모니터)로 출력하기 위해 사용 자바 가상 기계를 종료할 때 사용	
String		문자열을 저장하고 여러가지 정보를 얻을 때 사용	
Wrapper	Byte, Short, Character, Integer, Float, Double, Boolean, Long	기본타입의 데이터를 갖는 객체를 만들 때 사용 문자열을 기본타입으로 변환할 때 사용 입력값 검사에 사용	
Math		수학 함수를 이용할 때 사용	



## Object 클래스

클래스를 선언할 때 extends 키워드로 다른 클래스를 상속하지 않으면 java.lang.Object 클래스를 상속하게 된다.

자바의 모든 클래스는 Object 클래스를 상속받기 때문에 Object 클래스의 멤버를 자연스럽게 사용할 수 있다.





# 객체 비교(equals())

equals()메소드의 매개 타입은 Object이다.

Object클래스의 equals()메소드는 == 와 동일한 결과를 리턴한다.

```
Object obj1 = new Object();
Object obj2 = new Object();
boolean result1 = obj1.equals(obj2);
boolean result2 = (obj1 == obj2);
```



# 객체 비교(equals())

String 객체의 equals()메소드는 문자열을 비교하여 동일하다면 true를 반환 그렇지 않다면 false를 반환한다. 이것이 가능한 이유는 String 클래스가 Object의 equals()메소드를 재정의해서 번지 비교가 아닌 문자열 비교로 변경했기 때문이다.

```
String str1 = new String();
String str2 = new String();
boolean result1 = str1.equals(str2);
boolean result2 = (str1 == str2);
```



## equals()와 hashCode()

객체들를 자료구조 안에 저장을 할 때 자료구조 안에 들어가는 객체들은 두 객체의 데이터가 같은지 확인 후에 자료구조 안에 저장하는 것이 바람직하다. 이를 위해서 java.lang.Object의 equals()와 hashCode()메소드가 사용된다.

equals(): Override 해서 데이터가 같을때 true를 반환하도록 재정의를 한다. hashCode(): hashCode는 객체의 주소값을 변환한 객체의 고유한 정수값이다. 객체의 데이터가 같을 때 같은 hashCode를 반환하도록 재정의를 한다.





# 객체 문자정보(toString())

Object 클래스의 toString() 메소드는 객체의 문자 정보를 리턴한다. 객체의 문자정보란 객체를 문자열로 표현한 값이다. 기본적으로 Object 클래스의 toString()메소드는 '클래스이름@16진수해시코드'로 구성된 문자 정보를 리턴한다.

```
Object obj = new Object();
System.out.println(obj.toString());
```

java.lang.Object@1175e2db

Object클래스의 toString()메소드는 가치가 없는 정보이므로 하위클래스가 가치있는 정보로 메소드를 재정의한다. String 클래스의 toString()는 문자열을 그대로 반환하고, Wrapper 클래스의 toString() 메소드는 기본값을 문자열로 반환한다.



### System 클래스

자바 프로그램은 운영체제에서 바로 실행되는 것이 아니라 JVM 위에서 실행된다. 따라서 운영체제의 모든 기능을 직접 이용하기 어렵다. 하지만 System 클래스를 사용하면 운영체제의 일부 기능을 사용할 수 있다.

키보드로부터 입력, 모니터로 출력, 현재 시간 읽기 등이 가능하다.
System 클래스의 모든 필드와 메소드는 정적필드 / 정적 메소드로 구성되어 있다.





#### System 클래스

System 클래스의 currentTimeMillis() 와 nanoTime() 메소드는 컴퓨터의 시계로부터 현재시간을 읽어서 밀리세컨드와 나노세컨드 단위의 long값을 리턴한다.

현재시간 읽기 → currentTimeMillis() / nanoTime()

- ex) long time = System.currentTimeMillis();
- ex) long time = System.nanoTime();



#### Runtime 클래스

Runtime 클래스는 자바 프로그램의 실행 환경을 나타내는 클래스이다. 자바 가상 머신(JVM)과 상호작용하여 프로그램을 실행하고 제어하는 기능을 제공한다.

Runtime 클래스는 자바 프로그램이 실행되는 동안에만 유효하며, JVM이 종료되면 함께 소멸된다. 프로그램의 실행 환경과 관련된 다양한 기능을 제공하여 프로그램의 실행 및 관리를 용이하게 해주는 클래스이다.





# Runtime 클래스

메소드	설명	
exec	명령어나 실행 파일을 실행 해당 프로세스와의 통신을 위한 입출력 스트림을 제공.	
totalMemory	JVM이 프로세스에 할당한 모든 메모리 영역의 크기를 합한 값을 반환	
freeMemory	JVM이 현재 사용하지 않는 메모리의 크기를 바이트 단위로 반환	
maxMemory	JVM이 최대로 할당 가능한 메모리 크기를 바이트 단위로 반환	
gc	가비지 컬렉션(Garbage Collection)을 수동으로 실행	
exit	자바 프로그램을 종료	





메소드	설명	예제코드	리턴값
int abs(int a) double abs(double a)	절대값	int $v1 = Math.abs(-5);$ double $v2 = Math.abs(3.14);$	v1 = 5 v2 = 3.14
double ceil(double a)	올림값	double v3 = Math.ceil(5.3); double v4 = Math.ceil(-5.3);	v3 = 6.0 v4 = -5.0
double floor(double a)	버림값	double v5 = Math.floor(5.3); double v6 = Math.floor(-5.3);	v5 = 5.0 v6 = -6.0
int max(int a, int b) double max(double a, double b)	최대값	int $v7 = Math.max(5, 9);$ double $v8 = Math.max(5.3, 2.5);$	v7 = 9 v8 = 5.3
int min(int a, int b) double min(double a, double b)	최소값	int $v9 = Math.min(5, 9);$ double $v10 = Math.min(5.3, 2.5);$	v9 = 5 v10 = 2.5
double random()	랜덤값	double v11 = Math.random();	0.0 <= v11 < 1.0
long round(double a)	반올림값	long v14 = Math.round(5.3); long v15 = Math.round(5.7);	v14 = 5 v15 = 6



# Arrays 클래스

메소드	설명	
static String toString(boolean[] v)	boolean 배열의 요소들를 문자열로 반환	
static String toString(byte[] v)	byte 배열의 요소들를 문자열로 반환	
static String toString(char[] v)	char 배열의 요소들를 문자열로 반환	
static String toString(short[] v)	short 배열의 요소들를 문자열로 반환	
static String toString(int[] v)	int 배열의 요소들를 문자열로 반환	
static String toString(long[] v)	long 배열의 요소들를 문자열로 반환	
static String toString(float[] v)	float 배열의 요소들를 문자열로 반환	
static String toString(double[] v)	double 배열의 요소들를 문자열로 반환	
static String toString(Object[] v)	Object 배열의 요소들를 문자열로 반환	



# Arrays 클래스

Ex) int[] arr =  $\{1, 2, 3, 4, 5\}$ ;

copyOf() / copyOfRange()	설명	
int[] arr2 = Arrays.copyOf(arr, arr.length);	arr 배열의 길이만큼 새로운 배열을 만들어 복사한다.	
int[] arr3 = Arrays.copyOf(arr, 3);	3길이만큼 새로운 배열을 만들어 길이만큼 복사한다.	
int[] arr4 = Arrays.copyOf(arr, 7);	7길이만큼 새로운 배열을 만들고 arr배열을 복사한다. 나머진 defaul값으로 채운다.	
int[] arr5 = Arrays.copyOfRange(arr, 2, 4);	Index 2부터 3까지(4불포함) 길이의 배열을 만들고 Index 2부터 3까지복사한다.	
int[] arr6 = Arrays.copyOfRange(arr, 0, 7);	Index 0부터 6까지(7불포함) 길이의 배열을 만들고 arr배열을 복사 후 나머진 defaul값으로 채운다.	



# Arrays 의 sort()메소드

Arrays의 sort() 메소드는 전달받은 배열의 모든 요소를 오름차순으로 정렬 매개변수로 정렬할 배열을 전달받으며, 전달받은 원본배열의 순서를 변경

```
int[] arr = {5, 3, 4, 1, 2};
```

Arrays.sort(arr);

System.out.println(Arrays.toString(arr));

결과 : [1, 2, 3, 4, 5]



## java.util (Date / Calendar)

Date 클래스는 날짜를 표현하는 클래스이다. Date는 객체간에 날짜 정보를 주고받을 때 매개변수나 리턴타입으로 주로 사용된다. 현재시각 Date객체는 다음과 같이 생성할 수 있다.

Date now = new Date();

Calendar 클래스는 달력을 표현한 클래스이다. 추상클래스이므로 new 연산자를 사용해서 인스턴스를 생성할 수 없다.

Calendar now = Calendar.getInstance();