

메서드 / 설명	예 제	결 과
String(String s) 주어진 문자열(s)을 갖는 String인스턴스를 생성한다.	String s = new String("Hello");	s = "Hello"
String(char[] value) 주어진 문자열(value)을 갖는 String인스턴스 를 생성한다.	char[] c = {'H','e','l','l','o'} String s = new String(c);	s = "Hello"
String(StringBuffer buf) StringBuffer인스턴스가 갖고 있는 문자열과 같은 내용의 String인스턴스를 생성한다.	StringBuffer sb = new StringBuffer("Hello"); String s = new String(sb);	s = "Hello"
char charAt(int index) 지정된 위치(index)에 있는 문자를 알려준다. (index는 0부터 시작)	String s = "Hello"; String n = "0123456"; char c = s.charAt(1); char c2 = n.charAt(1);	c = 'e' c2 = '1'
String concat(String str) 문자열(str)을 뒤에 덧붙인다.	String s = "Hello"; String s2 = s.concat(" World");	s2 = "Hello World"
boolean contains(CharSequence s) 지정된 문자열(s)이 포함되었는지 검사한다.	String s = "abcdefg"; boolean b = s.contains("bc");	b = true
boolean endsWith(String suffix) 지정된 문자열(suffix)로 끝나는지 검사한다.	String file = "Hello.txt"; boolean b = file.endsWith("txt");	b = true
boolean equals(Object obj) 매개변수로 받은 문자열(obj)과 String인스턴 스의 문자열을 비교한다. obj가 String이 아니거나 문자열이 다르면 false를 반환한다.	String s = "Hello"; boolean b = s.equals("Hello"); boolean b2 = s.equals("hello");	b = true b2 = false
boolean equalsIgnoreCase(String str) 문자열과 String인스턴스의 문자열을 대소문자 구분없이 비교한다.	String s = "Hello"; boolean b = s.equalsIgnoreCase("HELLO"); boolean b2 = s.equalsIgnoreCase("heLLo");	b = true b2 = true
int indexOf(int ch) 주어진 문자(ch)가 문자열에 존재하는지 확인하여 위치(index)를 알려준다. 못 찾으면 -1을 반환한다.(index는 0부터 시작)	String s = "Hello"; int idx1 = s.indexOf('o'); int idx2 = s.indexOf('k');	idx1 = 4 idx2 = -1

int indexOf(String str)	String s = "ABCDEFGH"; int idx = s.indexOf("CD");	idx = 2
주어진 문자열이 존재하는지 확인하여 그 위치(index)를 알려준다. 없으면 -1을 반환한다. (index는 0부터 시작)		
String intern()	String s = new String("abc"); String s2 = new String("abc"); boolean b = (s==s2); boolean b2 = s.equals(s2); boolean b3 = (s.intern()==s2.intern());	b = false b2 = true b3 = true
문자열을 constant pool에 등록한다. 이미 constant pool에 같은 내용의 문자열이 있을 경우 그 문자열의 주소값을 반환한다.		
int lastIndexOf(int ch)	String s = "java.lang.Object"; int idx1 = s.lastIndexOf('.'); int idx2 = s.indexOf('.');	idx1 = 9 idx2 = 4
지정된 문자 또는 문자코드를 문자열의 오른쪽 끝에서부터 찾아서 위치(index)를 알려준다. 못 찾으면 -1을 반환한다.		
int lastIndexOf(String str)	String s = "java.lang.java"; int idx1 = s.lastIndexOf("java"); int idx2 = s.indexOf("java");	idx1 = 10 idx2 = 0
지정된 문자열을 인스턴스의 문자열 끝에서 부터 찾아서 위치(index)를 알려준다. 못 찾으면 -1을 반환한다.		
int length()	String s = "Hello"; int length = s.length();	length = 5
문자열의 길이를 알려준다.		
String replace(char old, char nw)	String s = "Hello"; String s1 = s.replace('H', 'C');	s1 = "Cello"
문자열 중의 문자(old)를 새로운 문자(nw)로 바꾼 문자열을 반환한다.		
String replace(CharSequence old, CharSequence nw)	String s = "Hellollo"; String s1 = s.replace("ll", "LL");	s1 = "HeLLoLLo"
문자열 중의 문자열(old)을 새로운 문자열(nw)로 모두 바꾼 문자열을 반환한다.		
String replaceAll(String regex, String replacement)	String ab = "AABBAABB"; String r = ab.replaceAll("BB", "bb");	r = "AAbbAAbb"
문자열 중에서 지정된 문자열(regex)과 일치하는 것을 새로운 문자열(replacement)로 모두 변경한다.		
String replaceFirst(String regex, String replacement)	String ab = "AABBAABB"; String r = ab.replaceFirst("BB", "bb");	r = "AAbbAABB"
문자열 중에서 지정된 문자열(regex)과 일치 하는 것 중, 첫 번째 것만 새로운 문자열(replacement)로 변경한다.		

String[] split(String regex)	String animals = "dog,cat,bear"; String[] arr = animals.split(",");	arr[0] = "dog" arr[1] = "cat" arr[2] = "bear"
문자열을 지정된 분리자(regex)로 나누어 문자열 배열에 담아 반환한다.		
String[] split(String regex, int limit)	String animals = "dog,cat,bear"; String[] arr = animals.split(", ", 2);	arr[0] = "dog" arr[1] = "cat,bear"
문자열을 지정된 분리자(regex)로 나누어 문자열배열에 담아 반환한다. 단, 문자열 전체를 지정된 수(limit)로 자른다.		
boolean startsWith(String prefix)	String s = "java.lang.Object"; boolean b = s.startsWith("java"); boolean b2 = s.startsWith("lang");	b = true b2 = false
주어진 문자열(prefix)로 시작하는지 검사한다.		
String substring(int begin) String substring(int begin, int end)	String s = "java.lang.Object"; String c = s.substring(10); String p = s.substring(5, 9);	c = "Object" p = "lang"
주어진 시작위치(begin)부터 끝 위치(end) 범위에 포함된 문자열을 얻는다. 이 때, 시작위치의 문자는 범위에 포함되지만, 끝 위치의 문자는 포함되지 않는다.		
String toLowerCase()	String s = "Hello"; String s1 = s.toLowerCase();	s1 = "hello"
String인스턴스에 저장되어있는 모든 문자열을 소문자로 변환하여 반환한다.		
String toString()	String s = "Hello"; String s1 = s.toString();	s1 = "Hello"
String인스턴스에 저장되어 있는 문자열을 반환한다.		
String toUpperCase()	String s = "Hello"; String s1 = s.toUpperCase();	s1 = "HELLO"
String인스턴스에 저장되어있는 모든 문자열을 대문자로 변환하여 반환한다.		
String trim()	String s = " Hello World "; String s1 = s.trim();	s1 = "Hello World"
문자열의 왼쪽 끝과 오른쪽 끝에 있는 공백을 없앤 결과를 반환한다. 이 때 문자열 중간에 있는 공백은 제거되지 않는다.		
static String valueOf(boolean b) static String valueOf(char c) static String valueOf(int i) static String valueOf(long l) static String valueOf(float f) static String valueOf(double d) static String valueOf(Object o)	String b = String.valueOf(true); String c = String.valueOf('a'); String i = String.valueOf(100); String l = String.valueOf(100L); String f = String.valueOf(10f); String d = String.valueOf(10.0); java.util.Date dd = new java.util.Date(); String date = String.valueOf(dd);	b = "true" c = "a" i = "100" l = "100" f = "10.0" d = "10.0" date = "Sun Jan 27 21:26:29 KST 2008"
지정된 값을 문자열로 변환하여 반환한다. 참조변수의 경우, toString()을 호출한 결과를 반환한다.		