

String

클래스

```
char[ ] str = "문자열"
```



문자열을 조작하는 메서드

String클래스

Char배열에 기능(메서드)를 추가한 것이다.

Char배열과 그에 관련된 기능들을 함께 묶어서 클래스에 정의한다. 객체지향 개념이 나오기 이전의 언어들은 데이터와 기능을 따로 다루었지만 객체지향 언어에서는 데이터와 그와 관련된 기능을 하나의 클래스에 묶어서 다룰 수 있게 한다. 즉 서로 관련된 것들끼리 데이터와 기능을 함께 묶는 것이다.

String 객체 생성

```
//스트링 리터럴로 String 객체 생성
```

```
String str1 = "abcd"
```

```
//String 클래스의 생성자를 이용하여 String 객체 생성
```

```
char data[ ] = { 'a', 'b', 'c', 'd' };
```

```
String str2 = new String(data);
```

```
String str3 = new String("abcd");
```

생성자	설명
String()	빈 스트링 객체 생성
String(char[] value)	Char배열에 있는 문자들을 스트링 객체로 생성
String(String original)	매개변수로 주어진 문자열과 동일한 스트링 객체 생성

String클래스

주요 매서드

char charAt(int index)

int compareTo(String anotherString)

String concat(String str)

String[] split(String regex)

String trim()

String toUpperCase()

String toLowerCase()

String

method(메서드)

char [charAt](#)(int index)

int [length](#)()

[String](#)[] [split](#)([String](#) regex)

[String](#) [substring](#)(int beginIndex)

[String](#) [substring](#)(int beginIndex, int endIndex)

[String](#) [toUpperCase](#)()

[String](#) [toLowerCase](#)()

```
public class StringTest {  
  
    public static void main(String[] args) {  
  
        String s = "hello";  
        String s2= new String("hello");  
  
        System.out.println(s);  
  
        System.out.println(s.length());  
        System.out.println(s.charAt(0));  
        System.out.println(s.toUpperCase());  
        System.out.println(s.toLowerCase());  
  
        System.out.println(s.compareTo(s2)); // 같으면 0을 줌  
  
    }  
  
}
```

static 의 문자열상수 저장되는 공간

```
// 문자열상수 (스트링 리터럴로 String 객체생성)  
String str1="abcd";
```

new에 의해 heap공간에 개별적으로

```
//String 클래스의 생성자를 이용하여 String 객체 생성  
char data[] = {'a', 'b', 'c', 'd'};  
String str2 = new String(data);  
String str3 = new String("abcd");
```

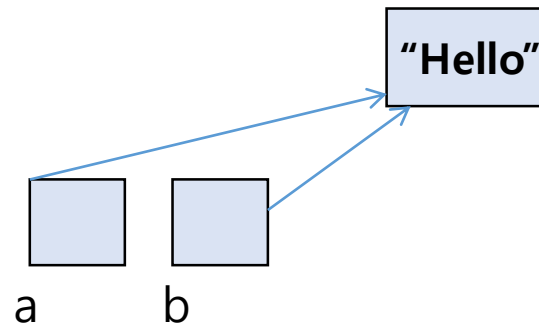
String

스트링리터럴(문자열상수)

```
String a="Hello";  
String b="Hello";
```

```
if( a==b)  "a와b의 참조값이 같다"  
if(a.equals(b))  "a와 b의 내용값이 같다"
```

정적static영역

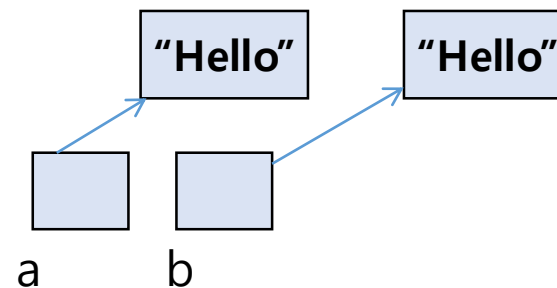


new String()

```
String c= new String("Hello");  
String d= new String("Hello");
```

```
if( c==d)  "c와d의 참조값이 같다"  
if(c.equals(d))  "c와 d의 내용값이 같다"
```

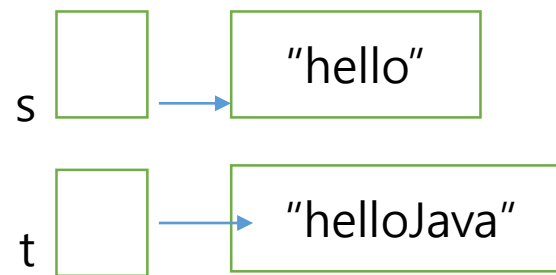
Heap영역



스트링 객체는 수정이 불가능하다

```
String s = new String("hello");  
String t = s.concat("java");
```

```
public class StringEx {  
  
    public static void main(String[] args) {  
        String s = new String("hello");  
        String t = s.concat("java");  
  
        System.out.println(s);  
        System.out.println(t);  
    }  
}
```



```
public class Test {  
  
    public static void main(String[] args) {  
  
        String date1 = new String("2018-02-19");  
        String[] s = date1.split("-");  
  
        for(int i=0; i<s.length; i++)  
        {  
            if( Integer.parseInt(s[i]) <10 && s[i].contains("0"))  
                s[i] = s[i].replace("0","");  
        }  
  
        System.out.println(s[0]+ "." + s[1] + "." + s[2]);  
    }  
}
```

```
public class Test {  
  
    public static void main(String[] args) {  
  
        String text = "Love is a variety of different feelings, states, and" +  
            "attitudes that ranges from interpersonal affection to pleasure";  
  
        int cnt=0;  
        for(int i=0; i<text.length(); i++)  
        {  
            if(text.charAt(i) == 'a')cnt++;  
        }  
        System.out.println("a문자 :" + cnt);  
    }  
}
```

```
public class StringMethod {  
  
    public static void main(String[] args)  
    {  
        String str="AWESOME-amazing";  
        // String str= new String("AWESOME-amazing");  
  
        System.out.println( "문자열의 길이 =" + str.length());  
        System.out.println("대문자로 변환="+ str.toUpperCase());  
        System.out.println("2번째 문자 출력" + str.charAt(2));  
        System.out.println("2번째 문자부터 부분만 가져오기"+ str.substring(2));  
        System.out.println("1번째 문자 이후부터 4번째 문자열전 까지 가져오기"+  
            str.substring(1,4));  
  
        String[] result = str.split("-");  
  
        for(String n : result)           //for each 구문  
            System.out.println(n);  
  
        //for(int i=0; i< result.length; i++) 위와 같은 결과  
        //      System.out.println( result[i]);  
    }  
}
```

String

사전상 등록순서에서 앞인지 뒤인지 같은지를
비교 판단

compareTo 메서드

문자열 정렬을 위한 비교(크다 ,작다 ,같다)

`int compareTo (String anotherString)`

내가 크면 양수, 내가 작으면 음수 같으면 0 반환

```
String name="apple";
```

```
String another="banana";
```

```
compareTo : 같으면 0, 내가 크면(사전 상 뒤에 오는 문자열이면) 양수  
내가 작으면 음수
```

```
public class StringCompareEx {  
  
    public static void main(String[] args) {  
  
        String name= new String("apple");  
        String anotherName = new String("banana");  
  
        System.out.println( name.compareTo(anotherName));  
  
        //문자열 정렬에 사용됨(사전 순 으로 먼저 나오면 작고 뒤에 나오면  
        //큰)  
        //현재문자열이 작으면 음수 값이 나옴    <0    //내림차순 (음수)  
        //현재문자열이 크면 양수 값이 나옴      >0    //오름차순 (양수)  
        //현재문자열과 비교문자열이 같으면 0 나옴  
    }  
}
```

```
public class StringCompare{
    public static void main(String[] args)
    {
        String tmp;
        String[] a = { "아이린", "하늬 ", "트와이스"};

        for(int i =0; i< a.length ; i++) //selection sort
        {
            for(int j=i+1 ; j< a.length ; j++)
            {
                if( a[i].compareTo(a[j]) >0 ) //오름차순
                {
                    tmp = a[i];
                    a[i]= a[j];
                    a[j]=tmp;
                }
            }
        }

        for(int i=0 ; i< a.length ; i++)
            System.out.println("a[i]="+ a[i]);
    }
}
```