

# String 클래스

# String 클래스

---

## ❖ String 생성자

- byte[] 배열을 문자열로 변환하는 생성자
  - 파일의 내용을 읽거나, 네트워크를 통해 받은 데이터 → byte[]

```
//배열 전체를 String 객체 생성
String str = new String(byte[] bytes);

//지정한 문자셋으로 디코딩
String str = new String(byte[] bytes, String charsetName);

//배열의 offset 인덱스 위치부터 length 개 만큼 String 객체 생성
String str = new String(byte[] bytes, int offset, int length);

//지정한 문자셋으로 디코딩
String str = new String(byte[] bytes, int offset, int length, String charsetName)
```

# String 클래스

---

## ❖ 바이트 배열을 문자열로 변환 : ByteToStringExample.java

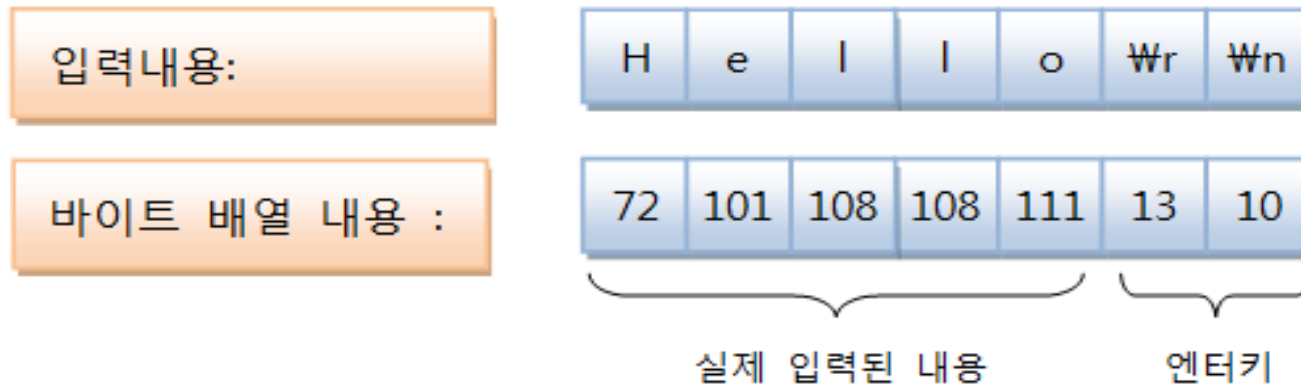
```
public class ByteToStringExample {  
    public static void main(String[] args) {  
        byte[] bytes = { 72, 101, 108, 108, 111, 32, 74, 97, 118, 97 };  
  
        String str1 = new String(bytes);  
        System.out.println(str1);  
  
        String str2 = new String(bytes, 6, 4);  
        System.out.println(str2);  
    }  
}
```

# String 클래스

## ❖ 생성자

- 키보드로부터 읽은 바이트 배열을 문자열로 변환

```
byte[] bytes = new byte[100];  
int readByteNo = System.in.read(bytes);  
String str = new String(bytes, 0, readByteNo-2);
```



# String 클래스

## ❖ 바이트 배열을 문자열로 변환 : KeyboardToStringExample.java

```
import java.io.IOException;

public class KeyboardToStringExample {
    public static void main(String[] args) throws IOException {
        byte[] bytes = new byte[100];    // 읽을 바이트를 저장하기 위한 배열 생성

        System.out.print( " 입력: ");
        // 배열에 읽은 바이트를 저장하고 읽은 바이트 수를 리턴
        int readByteNo = System.in.read(bytes);

        // 배열을 문자열로 변환
        String str = new String(bytes, 0, readByteNo - 2);
        System.out.println(str);
    }
}
```

# String 클래스

## ❖ 주요 메서드

리턴타입	메소드명(매개변수)	설명
char	charAt(int index)	특정 위치의 문자 리턴
boolean	equals(Object anObject)	두 문자열을 비교
byte[]	getBytes()	byte[]로 리턴
byte[]	getBytes(Charset charset)	주어진 문자셋으로 인코딩한 byte[]로 리턴
int	indexOf(String str)	문자열내에서 주어진 문자열의 위치를 리턴
int	length()	총 문자의 수를 리턴
String	replace(CharSequence target, CharSequence replacement)	target 부분을 replacement 로 대치한 새로운 문자열을 리턴
String	substring(int beginIndex)	beginIndex 위치에서 끝까지 잘라낸 새로운 문자열을 리턴
String	substring( int beginIndex, int endIndex)	beginIndex 위치에서 endIndex 전까지 잘라낸 새로운 문자열을 리턴
String	toLowerCase()	알파벳 소문자로 변환한 새로운 문자열을 리턴
String	toUpperCase()	알파벳 대문자로 변환한 새로운 문자열을 리턴
String	trim()	앞뒤 공백을 제거한 새로운 문자열을 리턴
String	valueOf(int i) valueOf(double d)	기본 타입값을 문자열로 리턴

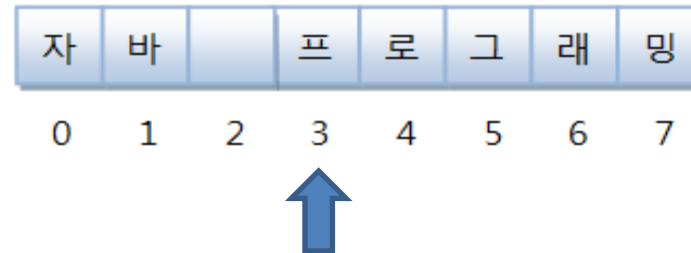
# String 클래스

---

## ❖ 문자 추출(charAt())

- 매개 값으로 주어진 인덱스의 문자 리턴

```
String subject = "자바 프로그래밍";  
char charValue = subject.charAt(3);
```



## String 클래스

---

### ❖ 주민등록번호에서 남자와 여자를 구분하는 방법: StringCharAtExample.java

```
public class StringCharAtExample {  
    public static void main(String[] args) {  
        String ssn = "010624-1230123";  
        char sex = ssn.charAt(7);  
        switch (sex) {  
            case '1':  
            case '3':  
                System.out.println("남자 입니다.");  
                break;  
            case '2':  
            case '4':  
                System.out.println("여자 입니다.");  
                break;  
        }  
    }  
}
```



# String 클래스

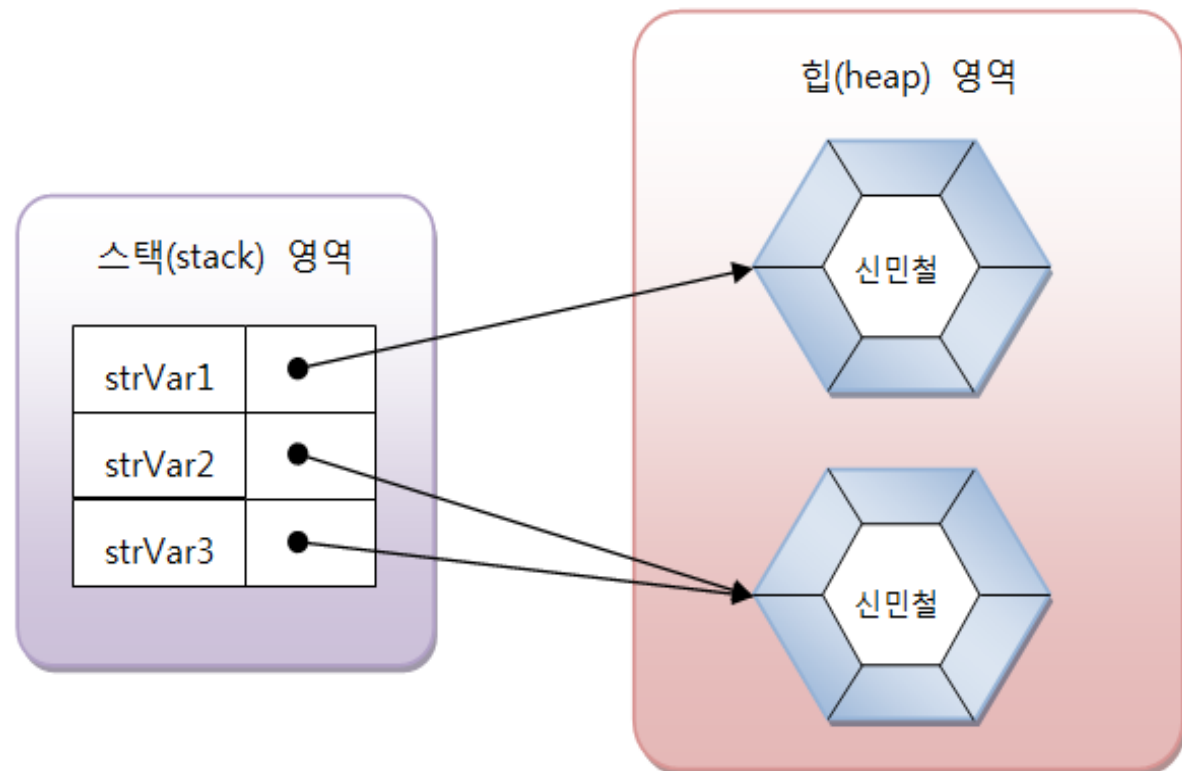
## ❖ 문자열 비교(equals())

- 문자열 비교할 때 == 연산자 사용하면 원하지 않는 결과 발생!

```
String strVar1 = new String("신민철");  
String strVar2 = "신민철";  
String strVar3 = "신민철";
```

```
strVar1 == strVar2 → false  
strVar2 == strVar3 → true
```

```
strVar1.equals(strVar2) → true  
strVar2.equals(strVar3) → true
```



# String 클래스

## ❖ StringEqualsExample.java

```
public class StringEqualsExample {  
    public static void main(String[] args) {  
        String strVar1 = new String("신민철");  
        String strVar2 = "신민철";  
  
        if (strVar1 == strVar2) {  
            System.out.println("같은 String 객체를 참조");  
        } else {  
            System.out.println("다른 String 객체를 참조");  
        }  
  
        if (strVar1.equals(strVar2)) {  
            System.out.println("같은 문자열을 가짐");  
        } else {  
            System.out.println("다른 문자열을 가짐");  
        }  
    }  
}
```

## String 클래스

---

### ❖ 바이트 배열로 변환(getBytes())

- 시스템의 기본 문자셋으로 인코딩된 바이트 배열 얻기

```
byte[] bytes = "문자열".getBytes();
```

- 특정 문자셋으로 인코딩 된 바이트 배열 얻기

```
try {  
    byte[] bytes = "문자열".getBytes("EUC-KR");  
    byte[] bytes = "문자열".getBytes("UTF-8");  
} catch (UnsupportedEncodingException e) {  
}
```

- [참고] 디코딩

```
String str = new String(byte[] bytes, String charsetName);
```

# String 클래스

---

## ❖ StringGetBytesExample.java

```
import java.io.UnsupportedEncodingException;

public class StringGetBytesExample {
    public static void main(String[] args) {
        String str = "안녕하세요";

        byte[] bytes1 = str.getBytes();
        System.out.println("bytes1.length: " + bytes1.length);
        String str1 = new String(bytes1);
        System.out.println("bytes1->String: " + str1);
    }
}
```

# String 클래스

---

## ❖ StringGetBytesExample.java

```
try {
    byte[] bytes2 = str.getBytes("EUC-KR");
    System.out.println("bytes2.length: " + bytes2.length);
    String str2 = new String(bytes2, "EUC-KR");
    System.out.println("bytes2->String: " + str2);

    byte[] bytes3 = str.getBytes("UTF-8");
    System.out.println("bytes3.length: " + bytes3.length);
    String str3 = new String(bytes3, "UTF-8");
    System.out.println("bytes3->String: " + str3);

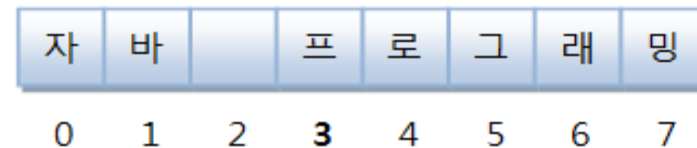
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
}
```

# String 클래스

## ❖ 문자열 찾기(indexOf(), lastIndexOf())

- 매개값으로 주어진 문자열이 시작되는 인덱스 리턴
- 주어진 문자열이 포함되어 있지 않으면 -1 리턴

```
String subject = "자바 프로그래밍";  
int index = subject.indexOf("프로그래밍");
```

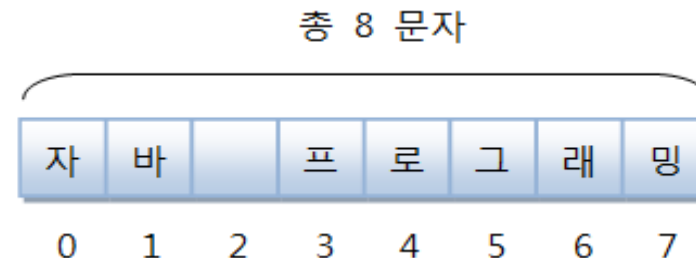


- 특정 문자열이 포함되어 있는지 여부에 따라 실행 코드 달리할 때 사용
- lastIndexOf()
  - 문자열 끝에서 부터 검색

## ❖ 문자열 길이(length())

- 공백도 문자에 포함

```
String subject = "자바 프로그래밍";  
int length = subject.length();
```



## String 클래스

---

### ❖ 문자열의 문자 수 얻기 : StringLengthExample.java

```
public class StringLengthExample {  
    public static void main(String[] args) {  
  
        String ssn = "7306241230123";  
        int length = ssn.length();  
  
        if (length == 13) {  
            System.out.println("주민번호 자리수가 맞습니다.");  
        } else {  
            System.out.println("주민번호 자리수가 틀립니다.");  
        }  
    }  
}
```

## String 클래스

---

### ❖ 문자열 포함 여부 조사 : StringIndexOfExample.java

```
public class StringIndexOfExample {  
  
    public static void main(String[] args) {  
  
        String subject = "자바 프로그래밍";  
  
        int location = subject.indexOf("프로그래밍");  
        System.out.println(location);  
  
        if (subject.indexOf("자바") != -1) {  
            System.out.println("자바와 관련된 책이군요");  
        } else {  
            System.out.println("자바와 관련없는 책이군요");  
        }  
    }  
}
```



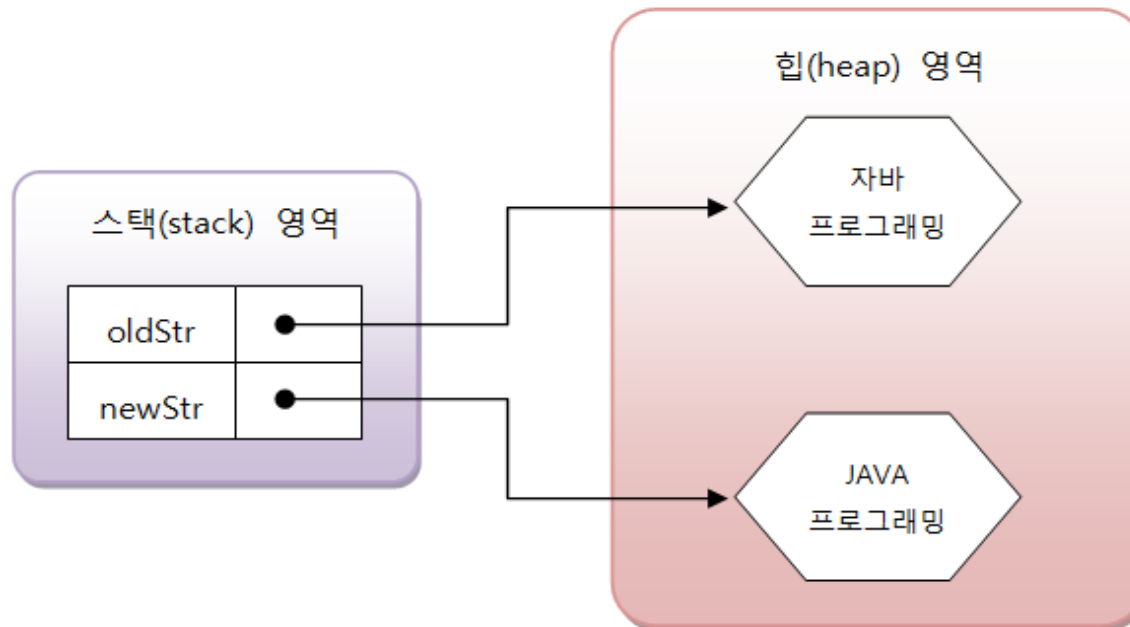
# String 클래스

## ❖ 문자열 대치(replace())

- 첫 번째 매개값인 문자열 찾을
- 두 번째 매개값인 문자열로 대치
- 새로운 문자열 리턴
  - 원본은 변화없음

```
String oldStr = "자바 프로그래밍";
```

```
String newStr = oldStr.replace("자바", "JAVA");
```



# String 클래스

---

## ❖ 문자열 대치하기: StringReplaceExample.java

```
public class StringReplaceExample {  
    public static void main(String[] args) {  
        String oldStr = "자바는 객체지향언어 입니다. 자바는 풍부한 API를 지원합니  
다.";  
        String newStr = oldStr.replace("자바", "JAVA");  
  
        System.out.println(oldStr);  
        System.out.println(newStr);  
    }  
}
```

# String 클래스

## ❖ 문자열 잘라내기(substring())

- substring(int beginIndex, int endIndex)
  - 주어진 시작과 끝 인덱스 사이의 문자열 추출
- substring(int beginIndex)
  - 주어진 인덱스 이후부터 끝까지 문자열 추출

8	8	0	8	1	5	-	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7	8	9	10	11	12	13

```
String ssn = "880815-1234567";  
String firstNum = ssn.substring(0, 6);  
String secondNum = ssn.substring(7);
```

# String 클래스

---

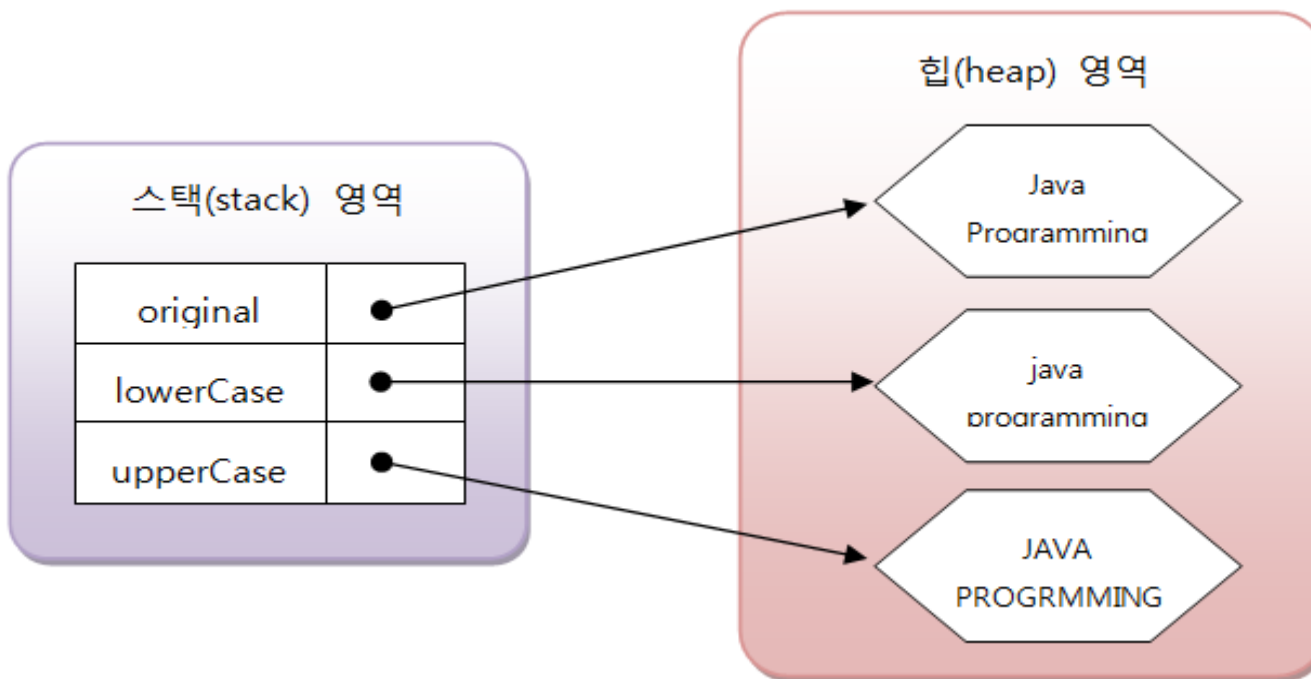
## ❖ 문자열 추출하기 : StringSubstringExample.java

```
public class StringSubstringExample {  
    public static void main(String[] args) {  
        String ssn = "880815-1234567 ";  
  
        String firstNum = ssn.substring(0, 6);  
        System.out.println(firstNum);  
  
        String secondNum = ssn.substring(7);  
        System.out.println(secondNum);  
    }  
}
```

# String 클래스

## ❖ 알파벳 소·대문자 변경 (toLowerCase(), toUpperCase())

```
String original = "Java Programming";  
String lowerCase = original.toLowerCase();  
String upperCase = original.toUpperCase();
```



# String 클래스

---

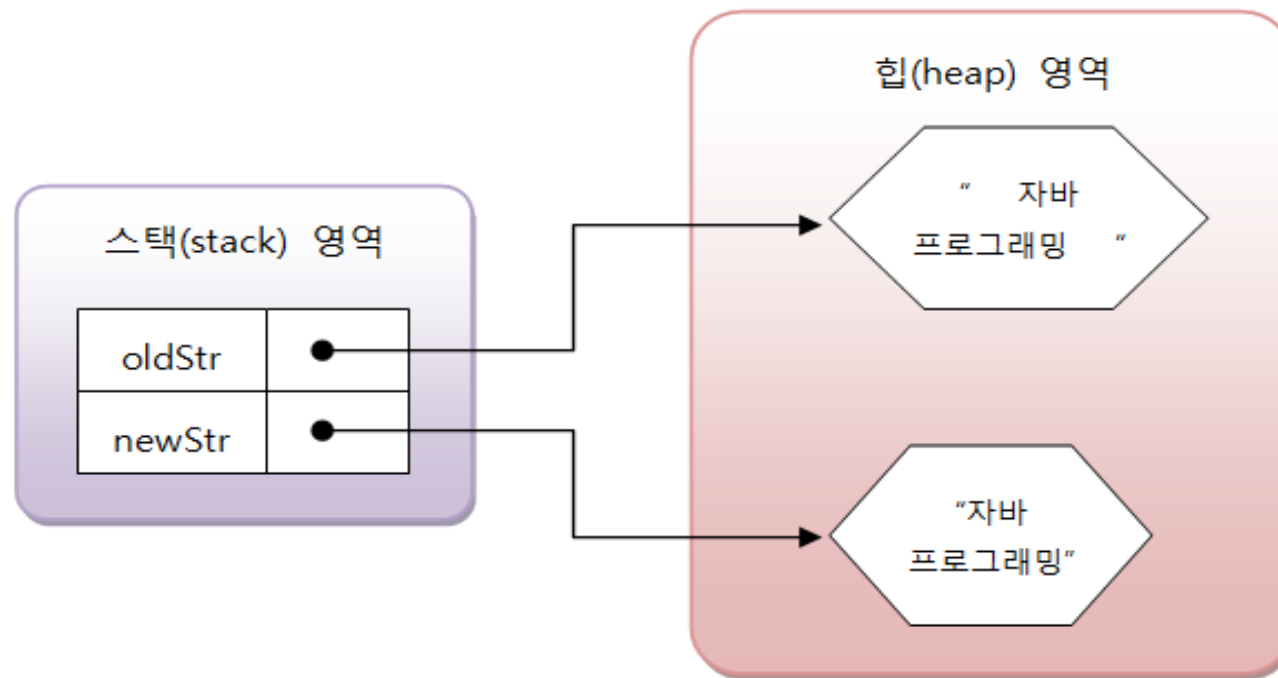
## ❖ 전부 소문자 또는 대문자로 변경 : StringToLowerUpperCaseExample.java

```
public class StringToLowerUpperCaseExample {  
    public static void main(String[] args) {  
        String str1 = "Java Programming";  
        String str2 = "JAVA Programming";  
  
        System.out.println(str1.equals(str2));  
  
        String lowerStr1 = str1.toLowerCase();  
        String lowerStr2 = str2.toLowerCase();  
        System.out.println(lowerStr1.equals(lowerStr2));  
  
        System.out.println(str1.equalsIgnoreCase(str2));  
    }  
}
```

# String 클래스

## ❖ 문자열 앞뒤 공백 잘라내기(trim())

```
String oldStr = " 자바 프로그래밍 ";  
String newStr = oldStr.trim();
```



# String 클래스

---

## ❖ 앞뒤 공백 제거 : StringTrimExample.java

```
public class StringTrimExample {  
    public static void main(String[] args) {  
        String tel1 = "  02";  
        String tel2 = "123  ";  
        String tel3 = "   1234   ";  
  
        String tel = tel1.trim() + tel2.trim() + tel3.trim();  
        System.out.println(tel);  
    }  
}
```



# String 클래스

---

## ❖ 문자열 변환(valueOf())

- 기본 타입의 값을 문자열로 변환

```
static String valueOf(boolean b)
static String valueOf(char c)
static String valueOf(int i)
static String valueOf(long l)
static String valueOf(double d)
static String valueOf(float f)
```

## String 클래스

---

### ❖ 기본 타입 값을 문자열로 변환: StringValueOfExample.java

```
public class StringValueOfExample {  
    public static void main(String[] args) {  
        String str1 = String.valueOf(10);  
        String str2 = String.valueOf(10.5);  
        String str3 = String.valueOf(true);  
  
        System.out.println(str1);  
        System.out.println(str2);  
        System.out.println(str3);  
    }  
}
```

# String 클래스

---

## ❖ 특정 문자열로 시작하는지, 끝나는지

- `boolean startsWith(String)`
  - URL에서 프로토콜 확인 및 서버 주소 확인시 사용 가능
- `boolean endsWith(String)`
  - 파일 확장명 판단시 사용 가능

# String 클래스

---

## ❖ StringWithExample.java

```
class StringWithExample {    //startsWith의 사용법
    public static void main(String args[]){
        String s="abc";
        boolean b1=s.startsWith("ab");
        boolean b2=s.startsWith("a");
        boolean b3=s.startsWith("bc");

        System.out.println(b1);
        System.out.println(b2);
        System.out.println(b3);
    }
}
```