commons-lang3

- description : commons-lang3 와 관련된 내용
- author : 주레피
- email : dhan@repia.com
- lastupdate : 2020-03-24

MavanRepository

```
<!-- https://mvnrepository.com/artifact/org.apache.commons/commons-lang3 -->
<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-lang3</artifactId>
    <version>3.8</version>
</dependency>
```

MVN Repository

StringEscapeUtils

- [ ] StringEscapeUtils (할당된 사용자 천호동밤안개. 마감일 2020-04-06)

StringUtils

> StringUtils.abbreviate (문자열, 표시할 문자 수)

긴 문자열을 원하는 만큼만 보여주고 나머지는 말줄임표로 생략해준다.

표시할 문자 수는 4 이상이어야 한다.
표시할 문자 수는 자동으로 붙는 말줄임표 … (3자) 를 포함한 수이다.

```
String abbreviate = StringUtils.abbreviate("ShowMeTheMoney", 7);
System.out.println(abbreviate);
//출력결과 : Show...
```

> StringUtils.appendIfMissing(문자열, 추가할단어, 마지막단어)

마지막단어가다를경우추가할단어를추가해준다
(대소문자를구분한다)

```
String appendIfMissing = StringUtils.appendIfMissing("ShowMeTheMoney", "-
```

```
From.Choi", "
Money");
System.out.println(appendIfMissing);
//출력결과 : ShowMeTheMoney
appendIfMissing = StringUtils.appendIfMissing("ShowMeTheMoney", "-
From.Choi", "test");
System.out.println(appendIfMissing);
//출력결과 : ShowMeTheMoney-From.Choi
```

> StringUtils.appendIfMissingIgnoreCase(문자열, “ 추가할단어 ”, “ 마지막단어 ”)

StringUtils.appendIfMissing과 같으나 대소문자를 구분하지 않는다.

```
String appendIfMissingIgnoreCase =
StringUtils.appendIfMissingIgnoreCase("ShowMeTheMoney", "-From.Choi",
"money");
System.out.println(appendIfMissingIgnoreCase);
//출력결과 : ShowMeTheMoney
```

> StringUtils.capitalize(문자열)

첫글자를 대문자로 변환한다.

```
String capitalize = StringUtils.capitalize("sss");
System.out.println(capitalize);
출력결과 : Sss
```

> StringUtils.chomp(문자열);

마지막에 개행문자 \n,\r,\r\n이 있을 경우 제거한다.

```
String chomp = StringUtils.chomp("ssssss\n");
System.out.println(chomp);
//출력결과 : ssssss
```

> StringUtils.chomp(문자열,제거할문자);

마지막에 제거할 문자가 있을 경우 제거한다.

```
String chomp2 = StringUtils.chomp("ssssss#","#");
System.out.println(chomp2);
//출력결과 : ssssss
```

> StringUtils.chop(문자열)

마지막 문자 하나를 제거한다.

```
String chop = StringUtils.chop("ssssss");
System.out.println(chop);
//출력결과 : sssss
```

> StringUtils.center(문자열, 글자수);

문자열에 설정한 글자수만큼 공백을 추가한다.
오른쪽, 왼쪽 순으로 추가된다.

```
String center = StringUtils.center("sss", 12);
System.out.println(center);
//출력결과 : sss
```

> StringUtils.center(문자열, 글자수, 추가될단어);

문자열에 설정한 글자수만큼 추가될 단어를 추가한다.
오른쪽, 왼쪽 순으로 추가된다.

```
String center2 = StringUtils.center("sss", 12, "#");
System.out.println(center2);
//출력결과 : ####sss#####
```

> StringUtils.compare(문자열, 비교할문자열)

문자열을 십진수로 변환후 비교한다.
문자를 유니코드 코드 (10진수) 로 변환 출력

```
System.out.println((int) '가');
System.out.println((int) '나');
//출력결과 : 44032
//출력결과 : 45208
int compare = StringUtils.compare("가", "나");
System.out.println(compare);
//출력결과 : -1176
//사용할 일이 있을려나...
```

> StringUtils.compareIgnoreCase(문자열, 비교할문자열)

문자열을 십진수로 변환후 비교한다.
(대소문자를 비교안함)

```
int compareIgnoreCase = StringUtils.compareIgnoreCase("S", "s");
System.out.println(compareIgnoreCase);
//출력결과 : 0
```

> StringUtils.contains(문자열, 비교할문자열)

문자열안에 비교할문자열이 포함되어 있으면
true를 반환한다.

```
boolean contains = StringUtils.contains("ssss", "ss");
System.out.println(contains);
//출력결과 : true
```

> StringUtils.containsAny(문자열, 비교할문자열, 비교할문자열)

문자열안에 비교할문자열이 하나라도 포함되어있으면
true를 반환한다.

```
boolean containsAny = StringUtils.containsAny("안녕하세요", "동안");
System.out.println(containsAny);
//출력결과 : true
```

> StringUtils.containsIgnoreCase(문자열, 비교할문자열)

문자열안에 비교할문자열이 포함되어있으면 true를 반환한다.
(대소문자를 비교안함)

```
boolean containsIgnoreCase = StringUtils.containsIgnoreCase("ss", "SS");
System.out.println(containsIgnoreCase);
//출력결과 : true
```

> StringUtils.containsNone(문자열, 비교할문자열)

문자열안에 비교할문자열이 하나라도
포함되어있으면 false를 반환한다.

```
boolean containsNone = StringUtils.containsNone("ssa", "axx");
System.out.println(containsNone);
//출력결과 : false
```

> StringUtils.containsOnly(문자열, 비교할문자열)

문자열안에 비교할문자열이 모두 포함되어있으면
true를 반환한다.

```
boolean containsOnly = StringUtils.containsOnly("ssa", "asd");
System.out.println(containsOnly);
//출력결과 : true
```

> StringUtils.containsWhitespace(문자열, 비교할문자열)

문자열안에 공백이 포함되어 있으면
true를 반환한다.

```
boolean containsWhitespace = StringUtils.containsWhitespace("sss sss sss");
System.out.println(containsWhitespace);
//출력결과 : true
```

> StringUtils.countMatches(문자열, 비교할문자열)

문자열안에 비교할 문자열이 있으면
출력결과에 1씩 추가한다.

```
int countMatches = StringUtils.countMatches("ccaaaa", "cc");
System.out.println(countMatches);
//출력결과 : 1
```

> StringUtils.defaultIfBlank(문자열, 문자열)

문자열이 띄어쓰기, 공백, null인 경우
반환할 문자열을 반환한다.

```
String defaultIfBlank = StringUtils.defaultIfBlank(" ", "띄어쓰기, 공백, null일시 반환문자");
System.out.println(defaultIfBlank);
//출력결과 : 공백일시 반환문자
```

> StringUtils.defaultIfEmpty(문자열, 반환할문자열)

문자열이 공백, null일 경우
'반환할 문자열'을 반환한다.

```
String defaultIfEmpty = StringUtils.defaultIfEmpty("", "공백,null일시 반환문
```

```
자");
System.out.println(defaultIfEmpty);
//출력결과 : 공백일시 반환문자
```

> StringUtils.defaultString(문자열, 반환할문자열)

문자열이 null인 경우
'반환할 문자열을 반환한다.
(반환할문자열이 없을경우 “ “ 로 반환한다.)

```
String defaultString = StringUtils.defaultString(null,"null일시 반환문자");
System.out.println(defaultString);
//출력결과 : null일시 반환문자
```

> StringUtils.deleteWhitespace(문자열)

문자열에 공백이 있을경우 제거한다.

```
String deleteWhitespace = StringUtils.deleteWhitespace("안녕 하세요");
System.out.println(deleteWhitespace);
// 출력결과 : 안녕하세요
```

> StringUtils.difference(문자열,비교할문자열)

문자열과 비교할문자열을 비교하고
비교할문자열에서 다른부분을 반환한다.

```
String difference = StringUtils.difference("안녕하시오", "안녕하세오");
System.out.println(difference);
//출력결과 : 세요
```

> StringUtils.endsWith(문자열,마지막문자열)

'문자열'의 마지막이 '마지막문자열'과 비교하고
같다면 true을 반환한다.

```
boolean endsWith = StringUtils.endsWith("안녕하세요.", ".");
System.out.println(endsWith);
//출력결과 : true
```

> StringUtils.endsWithAny(문자열,마지막문자열)

'문자열'의 마지막이 '마지막문자열'과 비교하고

같다면(vararg형식) true을 반환한다. (대/소문자를 구분한다)

```
boolean endsWithAny = StringUtils.endsWithAny("안녕하세요.s", ".S");
System.out.println(endsWithAny);
//출력결과 : false
```

> StringUtils.endsWithAny(문자열,마지막문자열)

문자열의 마지막이 마지막문자열과 비교하고
같다면 true을 반환한다.
(대/소문자를 구분안한다)

```
boolean endsWithIgnoreCase = StringUtils.endsWithIgnoreCase("안녕하세요.s",
".S");
System.out.println(endsWithIgnoreCase);
//출력결과 : true
```

> StringUtils.equals(문자열,비교할문자열)

문자열과 비교할문자열과 같다면
true을 반환한다.

```
boolean equals = StringUtils.equals("비교", "비교");
System.out.println(equals);
//출력결과 : true
```

> StringUtils.equalsAny(문자열,비교할문자열)

문자열과 비교할문자열과 같다면(vararg형식)
true을 반환한다.

```
boolean equalsAny = StringUtils.equalsAny("ss", "sS","ss");
System.out.println(equalsAny);
//출력결과 : false
```

> StringUtils.equalsAnyIgnoreCase(문자열,비교할문자열)

문자열과 비교할문자열과 같다면(vararg형식)
true을 반환한다.
(대/소문자를 구분안한다)

```
boolean equalsAnyIgnoreCase = StringUtils.equalsAnyIgnoreCase("ss",
"sS","ss");
```

```
System.out.println(equalsAnyIgnoreCase);
//출력결과 : true
```

> StringUtils.equalsIgnoreCase(문자열,비교할문자열)

문자열과 비교할문자열과 같다면
true을 반환한다.
(대/소문자를 구분안한다)

```
boolean equalsIgnoreCase = StringUtils.equalsIgnoreCase("ss", "sS");
System.out.println(equalsIgnoreCase);
//출력결과 : true
```

> StringUtils.firstNonBlank(문자열,문자열,문자열)

null, 공백, 띄어쓰기를 제외한 첫번째 문자열을 가져온다
StringUtils 3.8 버전부터 사용가능

```
String firstNonBlank = StringUtils.firstNonBlank(null, "", " ", "xyz");
System.out.println(firstNonBlank);
출력결과 : xyz
```

> StringUtils.getCommonPrefix(new String[] {“문자열 ”, “ 문자열 ”})

모든 문자열에서 공통된 첫부분을 출력한다.

```
String getCommonPrefix = StringUtils.getCommonPrefix("sss","ssS","ssa");
System.out.println(getCommonPrefix);
//출력결과 : ss
```

> StringUtils.getDigits(“문자열 ”)

Unicode Number(숫자)만 출력한다.

```
String getDigits = StringUtils.getDigits("123제거된다456");
System.out.println(getDigits);
//출력결과 : 123456
```

> StringUtils.indexOf(“문자열 ”, ” 찾을문자열 ”)

문자열중에 찾을 문자열이
포함된 첫번째 인덱스를 반환한다.

```
int indexOf = StringUtils.indexOf("sS", "S");
System.out.println(indexOf);
//출력결과 : 1
```

> StringUtils.indexOfAny(“문자열”,new String[] {“문자열 ” , “ 문자열 ” })

문자열과 배열을 비교하고(vararg형식)
첫번째 인덱스를 반환한다.

```
int indexOfAny = StringUtils.indexOfAny("안녕하세요",new String[] {"안", "녕"});
System.out.println(indexOfAny);
//출력결과 : 0
```

> StringUtils.indexOfAnyBut(“문자열”,new char[] {'문자', '문자'})

문자열과 문자 배열을 비교 후
배열에 없는 첫번째 문자인덱스를 반환한다.

```
int indexOfAnyBut = StringUtils.indexOfAnyBut("문자열",new char[] {'문', '자'});
System.out.println(indexOfAnyBut);
//출력결과 : 2
```

> StringUtils.indexOfDifference(“문자열 ” , “ 비교문자열 ”);

문자열과 비교문자열을 비교하고 달라지는
부분의 인덱스를 반환한다.

```
int indexOfDifference = StringUtils.indexOfDifference("안녕하세요","안녕하시
오");
System.out.println(indexOfDifference);
//출력결과 : 3
```

> StringUtils.indexOfIgnoreCase(“문자열 ” , “ 찾을문자열 ”)

문자열중에 찾을 문자열이 포함된 첫번째 인덱스를 반환한다.
(대/소문자를 구분 안한다)

```
int indexOfIgnoreCase = StringUtils.indexOfIgnoreCase("sS", "S");
System.out.println(indexOfIgnoreCase);
//출력결과 : 0
```

Ref

- StringUtils

주레피, eleven, stringutils, commonslang, commonslang3

From:
http://125.132.25.164/dokuwiki/ - **문제를 잘 정의하는 것은 문제를 절반 해결한 것이다. - 2023.12**

Permanent link:
http://125.132.25.164/dokuwiki/doku.php?id=wiki:java:commons-lang&rev=1586756554

Last update: **2022/03/10 19:52**