

String

String.prototype

Methods

static **String.array()** [{Array}](#)

 [AXUtil.js, line 1414](#)

콤마가 포함된 문자열을 Array로 변환합니다.

Example

```
1. var myObj = "a,b,c".array();  
2. trace(myObj);  
3. // ["a", "b", "c"]
```

static **String.crlf(replaceTarget, replacer)** [{String}](#)

 [AXUtil.js, line 1720](#)

줄넘김 문자열 '\n'을 >br/> 태그로 변환하여 줍니다.

Name	Type	Default	Description
replaceTarget	Regex	/\n/g	<small>optional</small>
replacer	String	>br/>	<small>optional</small>

Example

```
1. "123"  
2. "123".crlf(); -> "123<br/>123"
```

static **String.date(separator)** [{Date}](#)

 [AXUtil.js, line 1434](#)

문자열을 date 형식에 맞추어 날짜 포맷으로 리턴합니다.

Name	Type	Default	Description
separator	String	-	<small>optional</small> 날짜 짜 구분자

Example

```

1. trace("20121119".date());
2. // "2012-11-19T03:00:00Z"
3.
4. trace("2012-11-19".date());
5. // "2012-11-19T03:00:00Z"
6.
7. trace("2012/11/19".date("/"));
8. // "2012-11-19T03:00:00Z"

```

static **String.dec()** [{String}](#)

[AXUtil.js, line 1350](#)

URLencode된 문자열을 디코드 합니다.

Example

```

1. "AXJ_String%2C%EC%97%91%EC%8B%9C%EC%8A%A4%EC%A0%9C%EC%9D%B4".dec(); -> "AXJ_String,엑시스제이"

```

static **String.delHtml()** [{String}](#)

[AXUtil.js, line 1556](#)

문자열에서 HTML 태그를 제거하여 반환합니다.

Example

```

1. "<div>AXJ</div>".delHtml(); -> "AXJ"

```

static **String.delScript()** [{String}](#)

[AXUtil.js, line 1567](#)

문자열에서 Script 태그를 제거하여 반환합니다.

Example

```

1. "<script src='scriptname'></script>AXJ".delScript(); -> "AXJ"

```

static String.ecrlf(replaceTarget, replacer) [{String}](#)

[AXUtil.js, line 1734](#)

줄넘김 문자열 '%0A'을 >br/> 태그로 변환하여 줍니다.

Name	Type	Default	Description
replaceTarget	Regexp	/%0A/g	optional
replacer	String	>br/>	optional

Example

```
1. "123%0A123".ecrlf(); -> "123<br/>123"
```

static String.enc() [{String}](#)

[AXUtil.js, line 1375](#)

URLEncode된 문자열로 인코드 합니다.

Example

```
1. "AXJ_String,엑시스제이".enc(); ->
   "AXJ_String%2C%EC%97%91%EC%8B%9C%EC%8A%A4%EC%A0%9C%EC%9D%B4"
```

static String.getAnchorData() [{String}](#)

[AXUtil.js, line 1938](#)

anchor 데이터를 반환합니다.

Example

```
1. "http://jdoc.axisj.com/#{id:}/API/Prototype/String/phone\"}.getAnchorData();
2. "{id:}/API/Prototype/String/phone\"}"
```

static String.getBytes() [{Number}](#)

[AXUtil.js, line 1849](#)

문자열의 바이트 값을 계산하여 줍니다.

Example

```
1. trace("장".getBytes());
2. // 2
3. trace("a".getBytes());
```

4. `// 1`

static `String.getFileName()` [{String}](#)

 [AXUtil.js, line 1764](#)

파일경로에서 파일명을 반환합니다.

Example

```
1. "C://Works/AXISJ_project/css/myfile.zip".getFileName(); -> "myfile.zip"
```

static `String.lcase()` [{String}](#)

 [AXUtil.js, line 1827](#)

문자열을 소문자로 반환합니다.

Example

```
1. "AXISJ".lcase() -> "axisj"
```

static `String.left(strLen)` [{String}](#)

 [AXUtil.js, line 1324](#)

문자열 시작부터 지정한 글자수 만큼 반환합니다.

Name	Type	Description
<code>strLen</code>	Number	

Example

```
1. "AXJ_String".left(3); -> "AXJ"
2. toast.push('left(3) : ' + "AXJ_String".left(3));
```

static `String.money()` [{String}](#)

 [AXUtil.js, line 1813](#)

숫자형 문자열에 콤마를 삽입하여 통화단위로 반환합니다.

Example

```
1. "1000000".money()
2. // "1,000,000"
```

문자열을 Number로 변환해 줍니다.

Example

```
1. var str = "1234";
2. trace(typeof str);
3. // string
4.
5. str = str.number();
6. trace(typeof str);
7. // number
8.
9. "1,234".number(); -> 1234
10. "1,234.1".number(); -> 1234.1
```

JSONString이면 Object로 변환합니다.

Example

```
1. var myObj = "{a:1, b:2, name:'AXJ'}".object();
2. trace(myObj);
3. // {"a":1, "b":2, "name":"AXJ"}
4.
5. var myObjError = "{1, b:2, name:'AXJ'}".object();
6. trace(myObjError);
7. // {"error":"syntaxerr", "result":"syntaxerr", "msg":"JSON syntax error.{1, b:2,
```

```
name: 'AXJ' }", "body": "{1, b:2, name: 'AXJ' }"} }
```

static String.phone() [{String}](#)

 [AXUtil.js, line 1867](#)

문자열을 전화번호 형태로 반환합니다.

Example

```
1. trace("장".phone());
2. // 02
3. trace("a".phone());
4. // 02
5. trace("88819123".phone());
6. // 02-8881-9123
7. trace("01088819123".phone());
8. // 010-8881-9137
```

static String.queryToObject(separator) [{Object}](#)

 [AXUtil.js, line 1654](#)

queryString 형식의 문자열을 json object로 변환하여 줍니다.

Name	Type	Default	Description
separator	String	&	<div>optional</div>

Example

```
var myObject = "a=1&b=1".queryToObject();
trace(myObject);
// {"a": "1", "b": "1"}
```

static String.queryToObjectDec(separator) [{Object}](#)

 [AXUtil.js, line 1687](#)

queryString 형식의 문자열을 json object로 변환하여 줍니다. (파라미터 값은 URLDecode 합니다.)

Name	Type	Default	Description
separator	String	&	<div>optional</div>

Example

```
var myObject = "a=1&b=1".queryToObject();
trace(myObject);
// {"a":"1", "b":"1"}
```

static String.right(strLen) [{String}](#)

 [AXUtil.js, line 1337](#)

문자열 끝부터 지정한 글자수 만큼 반환합니다.

Name	Type	Description
strLen	Number	

Example

```
"AXJ_String".right(3); -> "ing"
toast.push('right(3) : '+$('#AXJrightTest').val().left(3));
```

static String.setDigit(length, padder) [{String}](#)

 [AXUtil.js, line 1747](#)

문자열 자리수를 맞추어 줍니다.

Name	Type	Default	Description
length	Number		
padder	String	0	<div>optional</div>

Example

```
"A".setDigit(3); -> "00A"
"A".setDigit(3, '!'); -> "!!A"
```

static String.times(count) [{String}](#)

 [AXUtil.js, line 1594](#)

문자열을 반복하여 반환합니다.

Name	Type	Description
count	Number	

Example

```
"AXJ".times(3); -> "AXJAXJAXJ"
```

Mozila 브라우저 등에서 사용하는 색상정보 값을 표준색상코드로 변환합니다. 표준색상코드를 입력하여도 표준색상코드 값을 얻을 수 있습니다.

Name	Type	Description
prefix	String	optional

Example

```
"rgb(243, 243, 243)".toColor(); -> "f3f3f3"
"rgb(243, 243, 243)".toColor('#'); -> "#f3f3f3"
"#f3f3f3".toColor(); -> "f3f3f3"
"f3f3f3".toColor(); -> "f3f3f3"
"f3f3f3".toColor('#'); -> "#f3f3f3"
```

문자열의 앞뒤 공백을 제거하여 줍니다.

Example

```
" AXJ ".trim(); -> "AXJ"
```

문자열을 대문자로 반환합니다.

Example

```
"axisj".ucase() -> "AXISJ"
```