

Number

Number.prototype

Methods

static **Number.abs()** [{Number}](#)

 [AXUtil.js, line 2151](#)

절대값을 반환합니다.

Example

```
1. trace((1234).abs());
2. // 1234
3. trace((-1234).abs());
4. // 1234
5. trace((1234.123).abs());
6. // 1234.123
7. trace((-1234.123).abs());
8. // 1234.123
```

static **Number.byte()** [{String}](#)

 [AXUtil.js, line 2071](#)

숫자값을 Byte로 인식하여 값에 크기에 따르 KB, MB, GB 의 형식으로 반환합니다.

Example

```
1. trace((1234567890).byte());
2. // 1.1GB
3. trace((12345678).byte());
4. // 11.8MB
5. trace((123456).byte());
```

```
6. // 120.6KB
```

```
7. trace((123).byte());
```

```
8. // 0.1KB
```

static **Number.ceil()** [{Number}](#)

 [AXUtil.js, line 2189](#)

Math.ceil

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

 [AXUtil.js, line 2203](#)

숫자를 time값으로 이용하여 Date를 반환합니다.

Example

```
1. var ndate = new Date();
```

```
2. ndate.getTime();
```

```
3. // 1417253161813
```

```
4. (1417253161813).date();
```

```
5. // Sat Nov 29 2014 18:26:01 GMT+0900 (KST)
```

static **Number.div(divisor)** [{Number}](#)

 [AXUtil.js, line 2218](#)

나누기 연산 결과를 반환합니다. divisor 가 0인 경우 연산 결과는 오류 없이 0을 반환합니다.

Name	Type	Description
divisor	Number	나눔수

Example

```
1. trace( (10).div(2); );
```

```
2. // 5
```

```
3. trace( (10).div(0); );
```

```
4. // 0
```

Math.floor□

숫자를 문자열로 변환하고 시작부터 지정한 글자수 만큼 반환합니다.

Name	Type	Description
strLen	Number	

Example

```
1. (1234).left(3); -> "123"
```

통화표현 단위로 변환된 문자열을 반환합니다.

Example

```
1. trace((1234.9).money());
```

```
2. //1,234.9
```

```
3. trace((1234.1).money());
```

```
4. //1,234.1
```

```
5. trace((-1234.9).money());
```

```
6. //-1,234.9
```

```
7. trace((-1234.1).money());
```

```
8. //-1,234.1
```

```
9.
```

```
10. (12345678).money(); -> "12,345,678"
```

```
11. "12345678".money(); -> "12,345,678"
```

```
12. // String 에서도 money 메소드를 직접 사용 할 수 있습니다.
```

static **Number.number()** [{Number}](#)[AXUtil.js, line 2101](#)

자신을 반환합니다.

static **Number.phone()** [{String}](#)[AXUtil.js, line 2244](#)

숫자를 문자로 변환후 String.phone를 실행합니다.

static **Number.rangeFrom(start)** [{Array}](#)[AXUtil.js, line 2132](#)

인자값부터 원본까지 정수 단위로 이어진 배열을 리턴합니다.

Name	Type	Description
start	Number	배열시작위치

Example

- `(3).rangeFrom(0);`
- `[0, 1, 2, 3]`

static **Number.right(strLen)** [{String}](#)[AXUtil.js, line 2018](#)

숫자를 문자열로 변환하고 마지막부터 지정한 글자수 만큼 반환합니다.

Name	Type	Description
strLen	Number	

Example

- `1234.right(3); -> 234`

static **Number.round(digit)** [{Number}](#)[AXUtil.js, line 2169](#)

반올림 위치에서부터 반올림 한 값을 반환합니다.

Name	Type	Description
digit	Number	

Example

```

1. trace((1234.5678).round());
2. //1235
3. trace((1234.5678).round(1));
4. //1234.6
5. trace((1234.5678).round(2));
6. //1234.57

```

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

 [AXUtil.js](#), line 2108

원하는 횟수 만큼 자릿수 맞춤 문자열을 포함한 문자열을 반환합니다.

Name	Type	Description
length	Number	자릿수
padder	String	자릿수 맞춤 문자열
radix	Number	진수

Example

```

1. trace( (11).setDigit(3) );
2. //011
3. trace( (11).setDigit(3, '!') );
4. //!11
5. trace( (11).setDigit(3, 0, 16) );
6. //00b
7. trace( (25).setDigit(5, "x", 8) );
8. //xxx31

```