



ZINCKSOFT

**Give The Dream New Life.**

*JSU.Misc*

*The complete Module API Reference*



<b>Quick Introduction</b>	<b>1</b>
Including the module in your page	1
<b>JSU.Misc API Reference</b>	<b>2</b>
<b>Properties</b>	<b>2</b>
<i>baseCssSize</i>	2
<b>Methods</b>	<b>2</b>
<i>parseCssUnit</i>	2
<i>getCssUnit</i>	2
<i>px2em / em2px</i>	3
<i>symbolicChmod / numericChmod</i>	3
<i>intRandom / floatRandom</i>	4

# Quick Introduction

The JSU Misc Module (**JSU.Color** / **JSUColor**) provides methods for miscellaneous functionality.

## Including the module in your page

Before including this module in your page, first include the core module:

```
<script type="text/javascript" src="jsu.core.js"></script>  
<script type="text/javascript" src="jsu.misc.js"></script>
```

# JSU.Misc API Reference

## A. Properties

### baseCssSize

This property is used for converting between pixels and ems. Its value defaults to 16, being the default for most modern browsers.

## B. Methods

### 1. parseCssUnit

Allows you to get the value of a CSS unit.

#### Syntax:

```
parseCssUnit(u)
```

#### Arguments:

- **u**: the css unit value - can be a pixel value, an em value, a plain number, and so on.

**Returns:** A float number.

#### Example:

```
alert(JSU.Misc.parseCssUnit("15")); // 15
alert(JSU.Misc.parseCssUnit("15px")); // 15
alert(JSU.Misc.parseCssUnit("1em")); // 1
alert(JSU.Misc.parseCssUnit("15%")); // 15
```

### 2. getCssUnit

Where parseCssUnit method would get the value of a CSS unit, this methods gets you the actual type of the specified CSS unit.

#### Syntax:

```
getCssUnit(u)
```

**Arguments:**

- **u**: the css unit value - can be a pixel value, an em value, a plain number, and so on.

**Returns:** A string.

**Example:**

```
alert(JSU.Misc.getCssUnit("15")); //
alert(JSU.Misc.getCssUnit("15px")); // px
alert(JSU.Misc.getCssUnit("1em")); // em
alert(JSU.Misc.getCssUnit("15%")); // %
```

**3. px2em / em2px**

These two methods allow you to convert between CSS pixel/em units.

**Syntax:**

```
px2em(x [, base [, decimals]])
em2px(x [, base])
```

**Arguments:**

- **x**: the css unit value;
- **base**: the browser's default font size used as a base for converting; defaults to 16.

**Returns:** px2em returns a float em value, and em2px returns an integer pixel value.

**Example:**

```
alert(JSU.Misc.px2em(12)); // 0.75
alert(JSU.Misc.px2em(12, 14)); // 0.857
alert(JSU.Misc.px2em(12, 16, 1)); // 0.8

alert(JSU.Misc.em2px(0.75)); // 12
alert(JSU.Misc.em2px(0.857, 14)); // 12
alert(JSU.Misc.em2px(0.8)); // 13
```

**4. symbolicChmod / numericChmod**

These methods allow you to convert unix permissions.

**Syntax:**

```
symbolicChmod(x [, isDirectory])
numericChmod(s);
```

**Arguments:**

- **x**: a number containing the value of the unix permissions;
- **s**: a string containing the unix formatted permissions;
- **isDirectory**: a boolean indicating whether to add the directory flag to the permissions.

**Returns:** symbolicChmod returns a string, numericChmod returns an integer.

**Example:**

```
alert(JSU.Misc.symbolicChmod(755)); // -rwxr-xr-x
alert(JSU.Misc.symbolicChmod(604, true)); // drw----r--

alert(JSU.Misc.numericChmod("drwxr-xr-x")); // 755
alert(JSU.Misc.numericChmod("-rw----r--")); // 604
```

**5. intRandom / floatRandom**

These methods are the enhanced versions of the standard JavaScript random method, allowing you to get a specific type and within a specific interval.

**Syntax:**

```
intRandom(h)
intRandom(l, h)
floatRandom(h)
floatRandom(l, h)
```

**Arguments:**

- **l**: the lower boundary (inclusive);
- **h**: the higher boundary (inclusive).

**Returns:** an integer or a float, depending on the chosen method.

**Example:**

```
alert(JSU.Misc.intRandom(-1,4)); // between [-1,4]
alert(JSU.Misc.intRandom(4)); // between [0,4]

alert(JSU.Misc.floatRandom(.5,.7)); // between [0.5,0.7]
alert(JSU.Misc.floatRandom(-1.1, 2.3)); // between [-1.1,2.3]
alert(JSU.Misc.floatRandom(.7)); // between [0,0.7]
```