



ZINCKSOFT

Give The Dream New Life.

JSU.Validation

The complete Module API Reference

Quick Introduction	1
Including the module in your page	1
JSU.Validation API Reference	2
Methods	2
<i>isValidUserName</i>	2
<i>isValidEmail</i>	2
<i>isValidHexColor</i>	3
<i>isValidIP / isValidIPv4 / isValidIPv6</i>	3
<i>isValidImageType</i>	4
<i>isValidTime12 / isValidTime24 / isValidTime</i>	4
<i>isValidDate</i>	5
<i>isValidPhoneNumber</i>	5
<i>isValidFloat / isValidFloatWithExp</i>	6
<i>isValidInteger</i>	6
<i>isValidCurrency</i>	7
<i>isEmptyLine / isWhiteLine</i>	7
<i>isValidHexNumber</i>	8
<i>isValidWindowsPath</i>	8
<i>isValidGUID</i>	9
<i>isValidSSN</i>	9
<i>Credit Card validation methods</i>	10
<i>isValidMAC</i>	11
<i>isValidLocale</i>	11
<i>isChar / isWord</i>	12
<i>onlyLetters / onlyDigits</i>	12
<i>isValidCssClass</i>	13
<i>isLeapYear</i>	13

Quick Introduction

The JSU Validation Module (**JSU.Validation** / **JSUValidaiton**) gives you the possibility to validate user input. It provides you with lots of validations methods for different needs, like credit card validation, email validation, ip (v4 and v6) validation, and so on.

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.validation.js"></script>
```

JSU.Validation API Reference

Methods

1. isValidUserName

Checks if a user name is valid. A valid value can contain letters, numbers, underscores, dots and dashes.

Syntax:

```
isValidUserName(name)
```

Arguments:

- **name**: the name to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidUserName("test.name")); // true
alert(JSU.Validation.isValidUserName("another/name")); // false
alert(JSU.Validation.isValidUserName("_hello-there")); // true
```

2. isValidEmail

Checks if an email address is valid.

Syntax:

```
isValidEmail(email)
```

Arguments:

- **email**: the email address to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidEmail("an@email.it")); // true
alert(JSU.Validation.isValidEmail("another@email.testing")); // true
alert(JSU.Validation.isValidEmail("hello@121.10.71.250")); // true
alert(JSU.Validation.isValidEmail("hi@hello")); // false
```

3. isValidHexColor

Checks if a string is a valid HEX formatted color. A valid hex color can start with the “#” symbol, followed by 3 or 6 hexadecimal digits.

Syntax:

```
isValidHexColor(color)
```

Arguments:

- **coloe**: the hex color to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidHexColor("fff")); // true
alert(JSU.Validation.isValidHexColor("#0f0f0f")); // true
alert(JSU.Validation.isValidHexColor("#gff010")); // false
```

4. isValidIP / isValidIPv4 / isValidIPv6

Checks whether an IP address is a valid IPv4 or IPv6 address.

Syntax:

```
isValidIP(ip)    alias: isValidIPv4(ip)
isValidIPv6(ip)
```

Arguments:

- **ip**: the ip address to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidIP("127.0.0.1")); // true
alert(JSU.Validation.isValidIP("0.0.0.0")); // true
alert(JSU.Validation.isValidIP("255.255.255.255")); // true
alert(JSU.Validation.isValidIP("1.0.0.256")); // false

alert(JSU.Validation.isValidIPv6(
    "2001:0db8:3c4d:0015:0000:0000:abcd:ef12")); // true
alert(JSU.Validation.isValidIPv6("2001:db8:85a3:8a2e:370:7334")); // false
```

5. isValidImageType

Checks whether a given filename is a valid image file. A valid filename contains an extension from the following list (case insensitive): JPEG, JPG, TIF, PNG, GIF, BMP.

Syntax:

```
isValidImageType(filename)
```

Arguments:

- **filename**: the filename to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidImageType("test.jpg")); // true
alert(JSU.Validation.isValidImageType("c:\\fo\\test.png")); // true
alert(JSU.Validation.isValidImageType("local//test.gif")); // true
alert(JSU.Validation.isValidImageType("testing")); // false
alert(JSU.Validation.isValidImageType("test.dot")); // false
alert(JSU.Validation.isValidImageType("test.jpX.png")); // true
```

6. isValidTime12 / isValidTime24 / isValidTime

Checks whether a given string represents a valid time value.

Syntax:

```
isValidTime12(time)
isValidTime24(time)
isValidTime(time [, hourFormat])
```

Arguments:

- **time**: the string to validate;
- **hourFormat**: an integer specifying the hour format (12 or 24); defaults to 12.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidTime("12:56")); // false
alert(JSU.Validation.isValidTime("12:56", 24)); // true
alert(JSU.Validation.isValidTime("1:23", 24)); // true
alert(JSU.Validation.isValidTime("4:27pm")); // true
alert(JSU.Validation.isValidTime("0:00 am")); // false
alert(JSU.Validation.isValidTime("12:00 am")); // true
```

7. isValidDate

Checks if a given string represents a valid date value. A valid date contains the individual date elements separated by a dash, space, slash, dot or a comma.

Syntax:

```
isValidDate(date [, format])
```

Arguments:

- **date**: the string to validate;
- **format**: the date format to validate against; defaults to DDMMYYYY, but can be one of the following:
 - DDMMYY / DDMMYYYY
 - MMDDYY / MMDDYYYY
 - YYMMDD / YYYYMMDD
 - YYDDMM / YYYYDDMM
 - DDMMYY / DDMMYYYY (three-letter month name)
 - MMDDYY / MMDDYYYY (three-letter month name)
 - YYMMDD / YYYYMMDD (three-letter month name)
 - YYDDMM / YYYYDDMM (three-letter month name)

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidDate("12/12/2004")); // true
alert(JSU.Validation.isValidDate("12/13/2006", "DDMMYYYY")); // false
alert(JSU.Validation.isValidDate("Jan. 13, 2004", "MMDDYYYY")); // true
alert(JSU.Validation.isValidDate("1/1,84", "DDMMYY")); // true
```

8. isValidPhoneNumber

Checks if a string represents a valid international phone number.

Syntax:

```
isValidPhoneNumber(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidPhoneNumber("(+900) 900-0121-1124")); // true
alert(JSU.Validation.isValidPhoneNumber("0700 400 500")); // true
alert(JSU.Validation.isValidPhoneNumber("121.200.354")); // false
```

9. isValidFloat / isValidFloatWithExp

Checks if a string represents a valid float value. The isValidFloatWithExp validates floats written in exponential notation.

Syntax:

```
isValidFloat(nr)
isValidFloatWithExp(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidFloat("-.256")); // true
alert(JSU.Validation.isValidFloat("12")); // true
alert(JSU.Validation.isValidFloat("123.0")); // true
alert(JSU.Validation.isValidFloat("12,5")); // true
alert(JSU.Validation.isValidFloat("12.5e-10")); // false

alert(JSU.Validation.isValidFloatWithExp("12.56")); // false
alert(JSU.Validation.isValidFloatWithExp("12.56e-12")); // true
```

10. isValidInteger

Checks if a string represents a valid integer value.

Syntax:

```
isValidInteger(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidInteger("12")); // true
alert(JSU.Validation.isValidInteger("-5")); // true
alert(JSU.Validation.isValidInteger("12.0")); // false
```

11. isValidCurrency

Checks if a string represents a valid currency value.

Syntax:

```
isValidCurrency(nr [, decimals])
```

Arguments:

- **nr**: the string to validate;
- **decimals**: the number of decimals allowed in the value; defaults to 2.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidCurrency("12,496,173.12")); // true
alert(JSU.Validation.isValidCurrency("12,496,173.12", 3)); // false
alert(JSU.Validation.isValidCurrency("12,121")); // true
alert(JSU.Validation.isValidCurrency("12.12")); // true
```

12. isEmptyLine / isWhiteLine

Checks if a line is empty or contains only white spaces or tabs.

Syntax:

```
isEmptyLine(s)
isWhiteLine(s)
```

Arguments:

- **s**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isEmptyLine("")); // true
alert(JSU.Validation.isEmptyLine("\n")); // false

alert(JSU.Validation.isWhiteLine("\t")); // true
alert(JSU.Validation.isWhiteLine("")); // false
alert(JSU.Validation.isWhiteLine(" ")); // true
```

13. isValidHexNumber

Checks if a string represents a hexadecimal number.

Syntax:

```
isValidHexNumber(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidHexNumber("fff")); // true
alert(JSU.Validation.isValidHexNumber("#0f0")); // false
alert(JSU.Validation.isValidHexNumber("a12g")); // false
alert(JSU.Validation.isValidHexNumber("0xffff")); // true
```

14. isValidWindowsPath

Checks if a given string represents a valid Windows path.

Syntax:

```
isValidWindowsPath(s)
```

Arguments:

- **s**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidWindowsPath("c:\\test")); // true
alert(JSU.Validation.isValidWindowsPath("c:\\\\test")); // false
alert(JSU.Validation.isValidWindowsPath("c:\\test\\")); // true
alert(JSU.Validation.isValidWindowsPath("c://test")); // false
alert(JSU.Validation.isValidWindowsPath("d:")); // true
alert(JSU.Validation.isValidWindowsPath("x:\\file.ext")); // true
```

15. isValidGUID

Checks if a string represents a valid Windows GUID value.

Syntax:

```
isValidGUID(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidGUID(
    "3F2504E0-4F89-11D3-9A0C-0305E82C3301")); // true
alert(JSU.Validation.isValidGUID(
    "{3F2504E0-4F89-11D3-9A0C-0305E82C3301}")); // true
alert(JSU.Validation.isValidGUID(
    "(3F2504E0-4F89-11D3-9A0C-0305E82C3301)")); // true
```

16. isValidSSN

Checks if a given string is a valid Social Security Number.

Syntax:

```
isValidSSN(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidSSN("127-45-1764")); // true
alert(JSU.Validation.isValidSSN("127451764")); // true
```

17. Credit Card validation methods

These methods allow you to check whether a credit card number is valid. The following credit cards are supported: VISA, MasterCard, AmericanExpress, DinersClub, Discover, JCB.

Syntax:

```
isValidVISA(nr)
isValidMasterCard(nr)
isValidAmericanExpress(nr)
isValidDinersClub(nr)
isValidDiscover(nr)
isValidJCB(nr)

isValidCreditCard(nr)
```

The **isValidCreditCard** method checks if a given string is a valid credit card number, regarding the credit card's type (as long as it's one of the supported credit cards).

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidVISA("411211121112111")); // true
alert(JSU.Validation.isValidVISA("411211121112")); // true
alert(JSU.Validation.isValidVISA("211211121112111")); // false

alert(JSU.Validation.isValidMasterCard("5212345678912345")); // true
alert(JSU.Validation.isValidMasterCard("5011111111111111")); // false

alert(JSU.Validation.isValidAmericanExpress("341234567891234")); // true
alert(JSU.Validation.isValidAmericanExpress("301234567891234")); // false

alert(JSU.Validation.isValidDinersClub("30212345678912")); // true
alert(JSU.Validation.isValidDinersClub("36212345678912")); // true
alert(JSU.Validation.isValidDinersClub("307111111111111")); // false
```

```

alert(JSU.Validation.isValidDiscover("6011123456789123")); // true
alert(JSU.Validation.isValidDiscover("6578123456789123")); // true
alert(JSU.Validation.isValidDiscover("6012111111111111")); // false

alert(JSU.Validation.isValidJCB("213112345678912")); // true
alert(JSU.Validation.isValidJCB("180012345678912")); // true
alert(JSU.Validation.isValidJCB("3512312345678912")); // true
alert(JSU.Validation.isValidJCB("7111111111111111")); // false

alert(JSU.Validation.isValidCreditCard("4111211121113")); // VISA: true

```

18. isValidMAC

Checks if a string represents a valid MAC network address.

Syntax:

```
isValidMAC(nr)
```

Arguments:

- **nr**: the string to validate.

Returns: A boolean.

Example:

```

alert(JSU.Validation.isValidMAC("00-0C-F1-56-98-AD")); // true
alert(JSU.Validation.isValidMAC("00:0C:F1:56:98:AD")); // true

```

19. isValidLocale

Checks if a given string represents a valid locale value. A valid value has the xx-XX or xx_XX format, where x/X is a letter.

Syntax:

```
isValidLocale(s)
```

Arguments:

- **s**: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isValidLocale("en-US")); // true
alert(JSU.Validation.isValidLocale("en_US")); // true
```

20. isChar / isWord

Checks if a string represents a valid character or word.

Syntax:

```
isChar(s)
isWord(s)
```

Arguments:

- *nr*: the string to validate.

Returns: A boolean.

Example:

```
alert(JSU.Validation.isChar("a")); // true
alert(JSU.Validation.isChar("")); // false
alert(JSU.Validation.isChar(" ")); // false
alert(JSU.Validation.isChar("\n")); // false
alert(JSU.Validation.isChar(":")); // true
alert(JSU.Validation.isChar("test")); // false

alert(JSU.Validation.isWord("test")); // true
alert(JSU.Validation.isWord("te_st")); // true
alert(JSU.Validation.isWord("test:")); // false
alert(JSU.Validation.isWord(" test ")); // false
alert(JSU.Validation.isWord("a test")); // false
```

21. onlyLetters / onlyDigits

Checks if a string contains only letters or digits.

Syntax:

```
onlyLetters(s)
onlyDigits(s)
```

Arguments:

- *s*: the string to validate.

Returns: A boolean.

Example:

```

alert(JSU.Validation.onlyLetters("asdae")); // true
alert(JSU.Validation.onlyLetters("asdae1")); // false

alert(JSU.Validation.onlyDigits("1214")); // true
alert(JSU.Validation.onlyDigits("-1210")); // false

```

22. isValidCssClass

Checks if a string represents a valid CSS class name. A valid value always starts with a letter, followed by other letters, numbers or underscore character.

Syntax:

```
isValidCssClass(c)
```

Arguments:

- *c*: the string to validate.

Returns: A boolean.

Example:

```

alert(JSU.Validation.isValidCssClass("aClass")); // true
alert(JSU.Validation.isValidCssClass("a_class")); // false
alert(JSU.Validation.isValidCssClass("a-class")); // true

```

23. isLeapYear

Checks if the given year is a leap year.

Syntax:

```
isLeapYear(year)
```

Arguments:

- *year*: a 2-digit or 4-digit integer.

Returns: A boolean.

Example:

```

alert(JSU.Validation.isLeapYear("2000")); // true
alert(JSU.Validation.isLeapYear("2002")); // false
alert(JSU.Validation.isLeapYear("84")); // true
alert(JSU.Validation.isLeapYear("1984")); // true

```