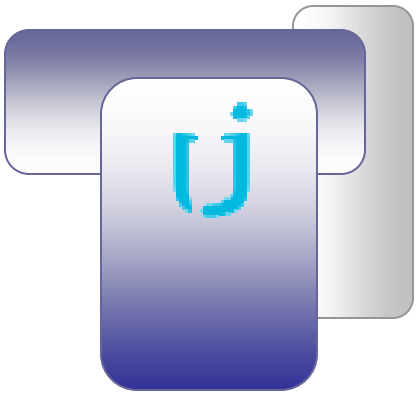

JeeUtil JS Library

Version 1.2 The Complete API Reference

JeeUtil 1.2: The Complete API Reference

Published Dec 2008. 1st edition for version v1.0

JeeUtil JavaScript Library[1] that aims to ease development of dynamic web applications. JeeUtil JS Library has been created by Jeevanandam and released as free software in May 2008. All content comes from the JeeUtil[1] site and is the intellectual property[2] of Jeevanandam.

[1] http://www.js.jeeutil.com
[2] http://www.js.jeeutil.com/license/jeeutil-ip.pdf

Table of Content

Chapter

1

1 How do I use JeeUtil JS Library on my web page?

JeeUtil JS Library is just like a **.js** file. It's very easy to use in your web page.

Please follow the below steps to use in your web page:

Step 1:

Download the JeeUtil JavaScript Library from http://js.jeeutil.com

Step 2:

Upload the .js file into your server (where you would like to keep it).

Step 3:

Uses below script tag include upload .js in your web page.
For Example: we are taking *www.js.jeeutil.com* as a server

Absolute Path-

```
<script type="text/javascript"
src="http://www.js.jeeutil.com/js/jeeutil-min-x.x. js"></script>
```

Relative Path-

```
<script type="text/javascript" src="/js/jeeutil-min-x.x.js">
</script>
```

Step 4:

Now you're ready utilize the features of JeeUtil JS Library in your web page.

Note:

x – Represents current version released in the http://js.jeeutil.com

Chapter

2

2 How do I use JeeUtil JS plug-in individually?

This approach will be of great use to those who need only partial features from JeeUtil JS Library. They can utilize this approach from JeeUtil.

JeeUtil JS plug-in is just like a **.js** file, it has part of the features from JS library. It's very easy to use on your web page.

Please follow the below steps to use in your web page:

Step 1:

Download the JeeUtil JavaScript Library – Plug-in from http://js.jeeutil.com
Eg: juCore-x.x.js, juBrowser-x.x.js, juDoc-x.x.js, juDateTime-x.x.js etc…

Step 2:

Upload the .js file into your server (where would you like keep).

Step 3:

Uses below script tag include upload .js in your web page.
For Example: we are taking *www.js.jeeutil.com* as a server

Absolute Path-

```
<script type="text/javascript"
src="http://www.js.jeeutil.com/js/juCore-min-x.x. js"></script>
```

Relative Path-

```
<script type="text/javascript" src="/js/juCore-min-x.x. js">
</script>
```

Step 4:

Now you're ready utilize the features of JeeUtil JS Library in your web page.

Note:

x – Represents current version released in the http://js.jeeutil.com

Chapter

3

3 juCore v1.0

---

ver

Returns version of juCore component in use.

```
alert(ju.ver);

// It will show 1.0.0

if(ju.ver.equalTo('1.0.0')){
        /* rest of the code goes here */
}
```

Example 3.1

---

isEqual

Checks whether given two objects are same data type or not.

Parameters
object-1
object-2

Return Value
True -> objects are same data type
False -> objects are not a same data type

```
if(ju.isEqual(object-1, object-2)){
        /* rest of the code goes here */
}
```

Example 3.2

isBool

Checks whether given object is Boolean or not.

Parameters
object

Return Value
True -> object is Boolean
False -> object is not a Boolean

```
if(ju.isBool(object)){
        /* rest of the code goes here */
}
```

Example 3.3

isFunction

Checks whether given object is a Function or not.

Parameters
object

Return Value
True -> object is Function
False -> object is not a Function

```
if(ju.isFunction(object)){
        /* rest of the code goes here */
}
```

Example 3.4

isList

Checks whether given object is List or not.

Parameters
object

Return Value
True -> object is List
False -> object is not a List

```
if(ju.isList(object)){
        /* rest of the code goes here */
}
```

Example 3.5

isNull

Checks whether given object or variable is Null or not.

Parameters
object or variable

Return Value
True -> object is Null
False -> object is not a Null

```
If(ju.isNull(object)){
        /* rest of the code goes here */
}
```

Example 3.6

isUndefined

Checks whether given object is undefined.

Parameters
object

Return Value
True -> object is undefined
False -> object is not an undefined

```
If(ju.isUndefined(object)){
        /* rest of the code goes here */
}
```

Example 3.7

isString

Checks whether given object or variable is String or not.

Parameters
Object or variable

Return Value
True -> object is String
False -> object is not a String

```
If(ju.isString(object)){
        alert(object);
        /* rest of the code goes here */
}
```

Example 3.8

addStylesheet

This API adds the external style sheet (CSS file) to the web page dynamically.

Parameters
filepath – absolute URL of the CSS file or relative path of CSS file

Return Value
True -> Style sheet added successfully in the web page
False -> Error occurred while adding style sheet in the web page

```
ju.addStylesheet("http://www.jeeutil.com/css/stylesheet.css");

ju.addStylesheet("/css/stylesheet.css");
```

Example 3.9

getPageBody

This API returns the current web page body object.

Parameters - None

Return Value
object -> returns web page body object
null -> no body tag present in the web page

```
ju.getPageBody( ); // it returns object
```

Example 3.10

focus

This API gives focus to almost all DOM HTML Elements in the web page.

Parameters
HTML-Element-Id

Return Value - None

```
<input type="text" id="username" name="username" size="15" />

ju.focus("username");                    // it gives focus to username text box
```

Example 3.11

getObj

Returns DOM[1] HTML Element from web page. The approach taken here is that it fetches an object anywhere in the page.
For example: you're within the iframe calling this API but element is residing outside iframe, still the function returns object from outside the iframe which gives you a greater flexibility.
It returns null when an object is not found, thus you can handle your code NOT TO EXECUTE when an element is not found instead of an error being thrown. This shall be specially useful when used with dynamic pages as the elements may or may not be generated as per the data available.

Parameters
HTML-Element-Id

Return Value
object -> DOM object based id passed
null -> given Element id is not found

```
<input type="password" id="password" size="20" />

ju.getObj("password");              //returns object
ju.getObj("sample");                //returns null, not element not found
```

Example 3.12

addFunction

This API gives freedom to add function to DOM HTML Element dynamically. It supports W3C[2] DOM events.

Parameters
HTML-Element-Id
Event Name
Function or Function Name

Return Value
True -> function added successfully
False -> error occurred while adding function
Null -> if given parameters is not valid

```
<input type="button" id="email_id" value="Show" />

//Assigning static (function name) function

function showValue( ){
        /* rest of the code goes here */
}

ju.addFunction("email_id", "click", showValue);    //adds the showValue to button
```

Example 3.13

[1] http://www.w3.org/TR/DOM-Level-2-Core/core.html#ID-getElBId
[2] http://www.w3schools.com/htmldom/dom_obj_event.asp

```
<input type=”button” id=”email_id” value=”Show” />

//Assigning dynamic function

ju.addFunction(“email_id”, “click”, function( ){

        /* rest of the code goes here */

        });
```

Example 3.14

---

deleteFunction

This API gives freedom to delete function from DOM HTML Element dynamically. It supports W3C[2] DOM events.

Note: this API supports only named functions.

Parameters
HTML-Element-Id
Event Name
Function

Return Value
True -> function deleted successfully
False -> error occurred while deleting function
Null -> if given parameters is not valid

```
<input type=”button” id=”email_id” value=”Show” />

//Deleting function from click event of button

function showValue( ){
        /* rest of the code goes here */
}

ju.deleteFunction(“email_id”, “click”, showValue);

//deletes the function click event from button
```

Example 3.15

replaceFunction

This API gives freedom to replace existing function from DOM HTML Element dynamically. It supports W3C[2] DOM events.

Note: this API supports only named functions.

Parameters
HTML-Element-Id
Event Name
Old-Function
New-Function

Return Value
True -> function replaced successfully
False -> error occurred while replacing function
Null -> if given parameters is not valid

```
<input type="button" id="email_id" value="Show" />

//Deleting function from click event of button

function showValue( ){
        /* rest of the code goes here */
}

function getValue( ){
        /* rest of the code goes here */
}

ju.replaceFunction("email_id", "click", showValue, getValue);

//it replaces the function click event in button
```

Example 3.16

getParentBackColor

API returns the back ground color of the parent DOM Element in the web page in the form of Hexadecimal color code.

Parameters
HTML-Element-Id

Return Value
Color-code -> returns color code in CSS eg.: #AA33DD
White-color-code -> returns when parent element doesn't have a back ground color.

```
<div style=”background-color:#33BDCF;”>
        <label id=”displayText”>Display Text</label>
</div>

ju.getParentBackColor(“displayText”);                // returns color code of parent
```

Example 3.17

---

getBackColor

API returns the back ground color of DOM Element in the web page. In the form of Hexadecimal color code.

Parameters
HTML-Element-Id

Return Value
Color-code -> returns color code in CSS eg.: #AA33DD
White-color-code -> returns when element doesn’t have a back ground color.

```
<div id=” displayText” style=”background-color:#33BDCF;”>
        <!-- rest of the HTML code goes here -->
</div>

ju.getBackColor(“displayText”);                // returns color code given Element ID
```

Example 3.18

---

getStyleProp

API returns the value of given CSS property from the HTML DOM Element.

Parameters
HTML-Element-Id

Return Value
prop-value -> returns given CSS property value
null -> returns if CSS property not set or not found in element

```
<div id=” displayText” style=”background-color:#33BDCF; width:300px;”>
        <!-- rest of the HTML code goes here -->
</div>

ju.getStyleProp(“displayText”, “width”);                // returns width of element
```

Example 3.19

fadeIn

This API gives entry appearance effect (fade) to the HTML Element in the page, this support all the HTML elements.

Parameters
HTML-Element-Id
Speed (optional) -> input range Between 1 to 15; Default is 5.

Return Value
null -> if element id is not valid

```
<img id="flower" src="/flower.jpg" alt="Flower" />

ju.fadeIn("flower");          // here it takes default speed value 5

ju.fadeIn("flower", 8);       // now it will take 8 as speed
```

Example 3.20

fadeOut

This API gives exit appearance effect (fade) to the HTML Element in the page, this support all the HTML elements.

Parameters
HTML-Element-Id
Speed (optional) -> input range Between 1 to 15; Default is 5.

Return Value
null -> if element id is not valid

```
<img id="flower" src="/flower.jpg" alt="Flower" />

ju.fadeOut("flower");         // here it takes default speed value 5

ju.fadeOut("flower", 8);      // now it will take 8 as speed
```

Example 3.21

3.1 Extending String Object

juCore enhances the String object with a series of useful API's for String.prototype ranging from the complex to the easy manipulation.

---

toInt

This API converts a string value into Integer form.

Parameters - None

Return Value
Integer-value -> Range of Integer

```
var value = "56536";
value.toInt( );                          //returns parsed Integer

//short hand usage
var value = "76766".toInt( );            //behavior same as above

var value = "6547657.7546".toInt( );     //return whole number this string
//-> 6547657
```

Example 3.1.1

---

toFloat

This API converts the string value into Floating form.

Parameters - None

Return Value
Float-value -> Range of Float[3]

```
var value = "56536.787";
value.toFloat();                         //returns parsed Integer

//short hand usage
var value = "76766.565".toFloat( );      //behavior same as above

//Integer to float conversion
var value = "65655".toFloat( );          //return Floating value with .00
//-> 65655.00
```

Example 3.1.2

[3] The JavaScript Number object does not allow you to set specific number types (like integer, short, long or double). In JavaScript all numbers are 64bit floating point numbers and have a number range from 5e-324 (negative) to 1.7976931348623157e+308 (positive).

equalTo

Determines whether this instance of String and a specified object (which must be a **String)**, have the same value.

Parameters
Object or String

Return Value
True -> both values are same.
False -> values are not same.

```
var str = “jeeutil!”;
str.equalTo(“JEEUTIL!”);                    //-> true

//short hand usage
var str = “JEEUTIL!”.equalTo(“Jeeutil!” );    //behavior same as above
```

Example 3.1.3

notEqualTo

Determines whether this instance of String and a specified object (which must be a **String)**, do not have the same value.

Parameters
Object or String

Return Value
True -> both values are not same.
False -> both values are same.

```
var str = “jeeutil!”;
str.notEqualTo(“JEEUTIL!”);                 //-> false

//short hand usage
var str = “JEEUTIL!”.notEqualTo(“Jeeutil!” );    //behavior same as above
```

Example 3.1.4

trim

Removes spaces on both the left and the right side of a string.

Parameters - None

Return Value
String

```
var str = “   jeeutil!   ”;
str.trim( );                                    //-> “jeeutil!”

//short hand usage
var str = “   jeeutil!   ”.trim( );             //behavior same as above
```

Example 3.1.5

---

rTrim

Removes spaces on the right side of a string.

Parameters - None

Return Value
String

```
var str = “   jeeutil!   ”;
str.rTrim( );                                   //-> “   jeeutil!”

//short hand usage
var str = “   jeeutil!   ”.rTrim( );            //behavior same as above
```

Example 3.1.6

---

lTrim

Removes spaces on the left side of a string.

Parameters - None

Return Value
String

```
var str = “   jeeutil!   ”;
str.rTrim( );                                           //-> “jeeutil!   ”

//short hand usage
var str = “   jeeutil!   ”.lTrim( );            //behavior same as above
```

Example 3.1.7

initCap

Capitalizes the first letter of a string and makes lowercase all the other letters.

Parameters - None

Return Value
String

```
var str = “jeeutil!”;
str.initCap( );                              //-> “Jeeutil!”

//short hand usage
var str = “JEEUTIL!”.initCap( );             //behavior same as above
```

Example 3.1.8

isEmpty

Checks, the string is empty or not. This trims the empty spaces prior to doing the check.

Parameters - None

Return Value
True -> string is empty
False -> string is not an empty

```
var str = “  ”;
str.isEmpty( );                          //-> true

//short hand usage
var str = “  ”.isEmpty( );               //behavior same as above

var str = “ jeeutil  ”.isEmpty( );       // false
```

Example 3.1.9

count

Returns the string length; excluding the white spaces at beginning and end.

Parameters - None

Return Value
count

```
var str = “  jeeutil  ”;
str.count( );                          //-> 7

//short hand usage
var str = “ jeeutil ”.count( );        //behavior same as above
```

Example 3.1.10

---

find

Checks whether a particular is string present in the object or not. This API is non-case sensitive. This is a much advanced version of the regex.

Parameters
Object or string

Return Value
True -> string found in the object
False -> string not found in the object

```
var str = “JeeUtil JavaScript Library”;
str.find(“library”);                                    //-> true

//short hand usage
var str = “JeeUtil JavaScript Library”.find(“library”);  //behavior same as above
```

Example 3.1.11

3.2 Extending Array Object

juCore enhances the Array object with a series of useful API's for Array.prototype ranging from the complex to the easy manipulation.

---

find

Checks whether object or string present in the Array or not. This API is non-case sensitive.

Parameters
Object or string

Return Value
Object -> found in the Array
False -> object not found in the Array

```
var str = [“hi”, “good”, “morning”, “thank”, “you”];
str.find(“Thank”);                                    //-> thank

//short hand usage
var str = [“hi”, “good”, “morning”, “thank”, “you”].find(“Thank”);

//behavior same as above
```

Example 3.2.1

---

isEmpty

Returns Array is empty or not; excluding empty Array indexes.

Parameters - None

Return Value
True -> Array is empty (with blank array value index in some position)
False -> Array contains objects or values

```
var str = [“”, “ ”, “  ”, “    ”];
str.isEmpty( );                                       //-> true

//short hand usage
var str = [“”, “ ”, “  ”, “    ”].isEmpty( );         //behavior same as above

var str = [“”, “ ”, “jeeutil”, “    ”].isEmpty( );     //-> false, it contains value
```

Example 3.2.1

Chapter

4

4 juBrowser v1.2

---

ver

Returns version of juBrowser plug-in in use.

Parameters - None

Return Value
String

```
alert(ju.browser.ver);

// It will show 1.0.1

if(ju.browser.ver.equalTo('1.0.1')){
        /* rest of the code goes here */
}
```

Example 4.1

name

Returns name of the browser currently used for surfing.

Browser Code	Browser Name
IE	Internet Explorer
FF	Firefox
GC	Google Chrome
SAF	Safari
OP	Opera
MOZ	Mozilla
NAV / NS	Netscape Navigator
AOL	America Online Browser
KONQ	Konqueror
SM	Sea Monkey
CAM	Camino
OW	Omni Web
FLK	Flock
MAX	Maxthon
GLN	Galeon
EPHY	Epiphany
KML	K-Meleon
PH	Phoenix
GDB	Gecko Debian
ICWS	Iceweasel
UKN	Unknown

*Note: Above list is in random order, *No Ranking has been taken into consideration.*

Parameters - None

Return Value
browser-code

```
If(ju.browser.name( ).equalTo('IE')){
        /* include IE specific action like changing CSS etc... */
}

If(ju.browser.name( ).equalTo('FF')){
        /* include Firefox specific action like above */
}
```

Example 4.2

version

Returns browser version of currently used for surfing.

Parameters - None

Return Value - In the form of name value pair
browser-version

```
var browserVersion = ju.browser.version( );

alert(browserVersion.major);                    //-> shows major version

alert(browserVersion.minor);                    //-> shows minor version
```

Example 4.3

isFPlayerExists

Checks Adobe Flash Player (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isFPlayerExists( )){
        alert("Adobe Flash Player is Installed.");
}
else{
        alert("Adobe Flash Player is not Installed");
}
```

Example 4.4

isAdobeReaderExists

Checks Adobe Acrobat Reader (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isAdobeReaderExists( )){
        alert("Adobe Reader is Installed.");
}
else{
        alert("Adobe Reader is not Installed");
}
```

Example 4.5

---

isRealPlayerExists

Checks Real Player (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isRealPlayerExists( )){
        alert("Real Player is Installed.");
}
else{
        alert("Real Player is not Installed");
}
```

Example 4.6

---

isWMPlayerExists

Checks Windows Media Player (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isWMPlayerExists( )){
        alert("Windows Media Player is Installed.");
}
else{
        alert("Windows Media Playe isr not Installed");
}
```

Example 4.7

isSilverlightExists

Checks Microsoft Sliverlight (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isSilverlightExists( )){
        alert("Microsoft Sliverlight is Installed.");
}
else{
        alert("Microsoft Sliverlight is not Installed");
}
```

Example 4.8

isQTPlayerExists

Checks Apple Quick Time Player (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isQTPlayerExists( )){
        alert("Apple Quick Time Player is Installed.");
}
else{
        alert("Apple Quick Time Player is not Installed");
}
```

Example 4.9

isDirectorExists

Checks Adobe Director (ActiveX & Plugin) is installed in the user system or not.

Parameters - None

Return Value
True -> Installed
False -> Not Installed

```
if(ju.browser.isDirectorExists( )){
        alert("Adobe Director is Installed.");
}
else{
        alert("Adobe Director is not Installed");
}
```

Example 4.10

---

isJava

Checks whether Java is enabled in the browser or not.

**** Kindly Note:** As you would know Java and JavaScript **are not the same**; they are two completely different and independent programming languages. The setting(s) for one does not affect the other.*

Parameters - None

Return Value
True -> Enabled
False -> Not enabled

```
if(ju.browser.isJava( )){
        /* rest of the code goes here */
}
```

Example 4.11

---

isCookie

Checks whether cookies are allowed in the browser or not.

Parameters - None

Return Value
True -> Cookies allowed
False -> Cookies not allowed

```
if(ju.browser.isCookie( )){
        /* rest of the code goes here */
}
```

Example 4.12

isCSS

Checks whether CSS is enabled in the browser or not.

Parameters - None

Return Value
True -> Enabled
False -> Not enabled

```
if(ju.browser.isCSS( )){
        /* rest of the code goes here */
}
```

Example 4.13

createCookie

This function helps to create a cookie for website while loading a page. By default cookie will expire in 1 day.

Parameters
*Cookie-name
*Value
No of days to Expires
Current path
Domain
Secure

* Mandatory

Return Value
True -> Cookie created successfully
Null -> Error occurred while creating cookie

```
ju.browser.createCookie(“jeeutil”, “username=<value>,otherinfo=<otherinfo
value>”, 2, “docs/browser”, “jeeutil.com”, true);

//-> cookie will be created with given values

ju.browser.createCookie(“jeeutil”, “<cookie-value>”);

//-> providing mandatory values only
```

Example 4.14

getCookie

Reads a cookie value from browser.

Parameters
Cookie-name

Return Value
Cookie-String -> cookie found
False -> not found
Null -> error occurred while reading

```
var cookieValue = ju.browser.getCookie("jeeutil");

//-> Returns cookie value
```

Example 4.15

getCookieLastUpdate

Returns the last updated time and date from cookie[1].

Parameters
Cookie-name

Return Value – In the Form of name value pair
Date-and-Time -> cookie found
False -> not found
Null -> error occurred while processing

```
var updateInfo = ju.browser.getCookieLastUpdate("jeeutil");
alert(updateInfo.date);          //-> shows date
alert(updateInfo.time);          //-> shows time
```

Example 4.16

[1] This API will return the value if cookie is created using JeeUtil JS Library

deleteCookie

Deletes the cookie from browser space.

Parameters
Cookie-name
Path
Domain

Return Value
True -> cookie deleted successfully
False -> not found
Null -> error occurred while deleting

```
if(ju.browser.deleteCookie("jeeutil", "docs/browser", "jeeutil.com")){
        /* rest of the code goes here */
}
```

Example 4.17

clearAllCookies

Deletes all cookies from browser against this website[2].

Parameters - None

Return Value
True -> cookies cleared successfully
False -> no cookies found
Null -> error occurred while clearing

```
var cookieValue = ju.browser.clearAllCookies( );

//-> Returns cookie value
```

Example 4.18

[2] Each browser has scope permission in cookie manipulation; so API *clearAllCookies* can able process against web site only. It will not clear all the cookies from browser because of browser Security.

popup

Creates a browser popup.

Parameters
*Popup-Name
*URL
Height
Width
Status
Toolbar
Menu-bar
Directories
Resizable
Scrollbars

*Mandatory
Rest all the parameter is optional and it will set it with default values.

Return Value
Popup-object-reference -> popup successfully created
False -> popup blocked/disabled in browser

```
var popObj = ju.browser.popup("jeeutil", "/docs/browser/index.htm", 600, 480);

if(popObj){
        /* rest of the code goes here */
}
```

Example 4.19

setStatus

Display the given string value in the browser status area[3].

Parameters
Display-String

Return Value - None

```
ju.browser.setStatus("Welcome to JeeUtil Library");

//-> it display in the browser status
```

Example 4.20

[3] In some of the browsers likes Firefox, Safari,etc, the option is controlled, so user need to enable this explicitly in browser.

language

Returns browser language, basically this is value user operating system language settings.

Parameters - None

Return Value
Language-Code[4]

```
var lang = ju.browser.language( );                    //-> for e.g: en-US, en-GB
```

Example 4.21

iWidth

Returns browser inner width.

Parameters - None

Return Value
Inner-width -> in pixels

```
var inner_width = ju.browser.iWidth( );              //->for eg: returns 548 (in pixels)
```

Example 4.22

iHeight

Returns browser inner height.

Parameters - None

Return Value
Inner-height -> in pixels

```
var inner_height = ju.browser.iHeight( );          //->for eg: returns 948 (in pixels)
```

Example 4.23

[4] http://www.js.jeeutil.com/?view=reference&id=langcode

scrollX

Returns browser X scrolling value.

Parameters - None

Return Value
x-scroll-value -> in pixels

```
var scroll_x = ju.browser.scrollX( );          //-> for eg: returns 30 (in pixels)
```

Example 4.24

scrollY

Returns browser Y scrolling value.

Parameters - None

Return Value
y-scroll-value -> in pixels

```
var scroll_y = ju.browser.scrollY( );          //-> for eg: returns 10 (in pixels)
```

Example 4.25

scrollXY

Returns browser X and Y scrolling Coordinates value. Coordinates

Parameters - None

Return Value – in the form name value pair
xy-coordinates -> in pixels

```
var coordinates = ju.browser.scrollXY( );

//-> returns Coordinates (in pixels)

coordinates.x                          //-> access X value
coordinates.y                          //-> access Y value
```

Example 4.26

Chapter

5

5 juDoc v1.1

---

ver

Returns version of juDoc plug-in in use.

Parameters - None

Return Value
String

```
alert(ju.doc.ver);

// It will show 1.0.0

if(ju.doc.ver.equalTo('1.0.0')){
        /* rest of the code goes here */
}
```

Example 5.1

setValue

Assigns a value to DOM Elements in the page; it supports almost all the DOM elements[1]. This is powered by W3C DOM manipulation[2].

Parameters
HTML-Element-Id
Value

Return Value
False -> given element Id or Value is not valid.

```
<div id="errorMsg" class="errorMsg"> </div>

ju.doc.setValue("errorMsg", "Given value is In-valid, please enter valid input.");

//-> it assigns value to DIV

<input type="text" id="email_Id" size="25" class="formInput" />

ju.doc.setValue("email_Id", "Please enter valid Email ID here.");

//-> it assigns value to INPUT element.
```

Example 5.2

getValue

Fetches a value from DOM Elements in the page; it supports almost all the DOM elements[1]. This is powered by W3C DOM manipulation[2].

Parameters
HTML-Element-Id

Return Value
Value -> element value; ***Note:*** for <select> if multiple select, it will return Array.
False -> given element Id.

```
<input type="text" id="username" size="15" class="formInput" />
<input type="password" id="email_Id" size="15" class="formInput" />

var username = ju.doc.getValue("username");
var password = ju.doc.getValue("password");

//-> it fetchs values from username and password (INPUT element).
```

Example 5.3

[1] http://www.js.jeeutil.com/?view=reference&id=elements
[2] http://www.w3.org/TR/DOM-Level-2-Core/core.html#ID-getElBId

isEmpty

Checks an HTML DOM element is empty or not.

Parameters
HTML-Element-Id

Return Value
True -> element value is empty.
False -> element value is not empty.

```
<input type="text" id="username" size="15" class="formInput" />

if(ju.doc.isEmpty("username")){
        alert("Username should not be empty, please enter.");
}
```

Example 5.4

isMailId

Checks an entered mail id is valid (syntax) format or not.

Parameters
HTML-Element-Id

Return Value
True -> mail id syntax is correct
False -> mail id syntax is not correct

```
<input type="text" id="email_id" size="25" class="formInput" />

if(!ju.doc.isMailId("email_id")){
        alert("Given mail id is not a valid form.");
}
```

Example 5.5

isNumeric

Checks a given element value is numeric (Numbers only) or not.

Parameters
HTML-Element-Id

Return Value
True -> valid numeric
False -> is not valid numeric

```
<input type="text" id="mobile_no" size="25" class="formInput" />

if(!ju.doc.isNumeric("mobile_no")){
        alert("Please enter valid numbers.");
}
```

Example 5.6

---

isInteger

Checks a given element value is Integer or not.

Parameters
HTML-Element-Id

Return Value
True -> valid Integer
False -> is not valid Integer

```
<input type="text" id="quantity" size="25" class="formInput" />

if(!ju.doc.isInteger("quantity")){
        alert("Please enter valid Quantity.");
}
```

Example 5.7

---

isReal

Checks a given element value is Real number or not.

Parameters
HTML-Element-Id

Return Value
True -> valid Real Number
False -> is not valid Real Number

```
<input type="text" id="average_score" size="25" class="formInput" />

if(!ju.doc.isReal("average_score")){
        alert("Please enter valid Average Score of your marks.");
}
```

Example 5.8

passCompare

Compares two password object values in the form.

Note: Input element type has to be a password.

Parameters
HTML-Element-Id

Return Value
True -> both password object value is equal
False -> password object value is not equal

```
<input type="password" id="pass" size="25" class="formInput" />
<input type="password" id="retype_pass" size="25" class="formInput" />

if(ju.doc.passCompare("pass", "retype_pass")){
        /* continue to get other inputs fro user */
}
else{
        alert("Entered password are not same.");
}
```

Example 5.9

validateForm

Checks the completeness of Form Elements.

Parameters
HTML-Element-Id's (comma separated values of ID)

Return Value
True -> All Form elements has value (based on passed Id's)
ElementID-Array -> these elements is empty

```
<input type="text" id="firstName" size="25" class="formInput" />
<input type="text" id="secondName" size="25" class="formInput" />
.
.
<input type="text" id="pincode" size="6" class="formInput" />

var idString = "firstName, secondName, .... , pincode";
var notEmptyElements = ju.doc.validateForm(idString);

if(notEmptyElements == true){
        /* user filled everything */
}
else{
        /* user left some input elements; rest code goes here */
}
```

Example 5.10

compareDate

Checks two date value. This API supports following date formats

- System - MM/DD/YYYY **or** MMDDYYY (Keyword is S)
- Custom - DD/MM/YYYY **or** DDMMYYYY (Keyword is C)

Parameters
*Date-value-1
*Date-value-2
Type

*Mandatory
Default “Type” value is System (S).

Return Value

<0 -> date1 is less than date2; returns difference between two date(s) in days

0 -> both equal

>0 -> date1 is greater than date2; returns difference between two date(s) in days

```
var date1 = “05/20/1984”;
var date2 = “07/21/1984”;

var result = ju.doc.compareDate(date1, date2);

if(result == 0){
        alert(“Given dates are equal”);
}
else if (result > 0){
        alert(“date2 is ” + result + “days greater than.”);
}
else if(result < 0){
        alert(“date2 is ” + result + “days less than.”);
}
```

Example 5.11

isAlpha

Checks a given element value is only Alphabets or not.

Parameters
HTML-Element-Id

Return Value
True -> valid Alphabets
False -> is not valid Alphabets

```
<input type="text" id="comment" size="25" class="formInput" />

if(!ju.doc.isAlpha("comment")){
        alert("Please enter valid comments in provided box.");
}
```

Example 5.12

Chapter

6

6 juShortcut v1.0

---

ver

Returns version of juShortcut plug-in being used.

Parameters - None

Return Value
String

```
alert(ju.shortcut.ver);

// It will show 1.0.0

if(ju.shortcut.ver.equalTo('1.0.0')){
        /* rest of the code goes here */
}
```

Example 6.1

add

This is a very powerful functuion that triggers functions based on shortcut keys in to an HTML element (Object).
This can be used to let user add a product to a shopping cart with a simple click like a Ctrl+A!
Or say if a user wants to add two items into the cart, they need to simply hit Ctrl+2 instead of the whole ‘But Button’ – ‘Add to cart’ – ‘Change number of item’ – ‘Update cart’ routine.

Parameters

*Key-Combinations	- eg: Ctrl+1, Ctrl+Alt+K
*Action	- function
Bubble	- Boolean
Disable-in-Input Element	- Boolean
Target-Element	- By default “document”

* Mandatory

Return Value - None

```
function addProduct( ){
        /* rest of the code goes here */
}

function deleteProduct( ){
        /*rest of the code goes here */
}

function showHelp( ){
        /* rest of the code goes here */
}


//Adding key shortcut dynamically in the web page

ju.shortcut.add(“Shift+A”, addProduct, true);

ju.shortcut.add(“Shift+D”, deleteProduct, true);

ju.shortcut.add(“Ctrl+F1”, showHelp, true);

//while user pressing above key combinations, respective function will be invoked.
```

Example 6.2

remove

Removes a previously added shortcut using add Method from HTML element (Object).

Parameters
Key-Combinations - eg: Ctrl+1, Ctrl+Alt+K

Return Value - None

```
/*
 * Please refer above add method Example 4.2
 */

//Removing key shortcut dynamically from the web page

ju.shortcut.remove("Ctrl+F1");



//above key combination will be removed from web page, while user pressing above key
//combinations nothing will happen (Respective function will not be called).
```

Example 6.3

Chapter 7

7 juDragdrop v1.0

---

ver

Returns version of juDragdrop plug-in in use.

Parameters - None

Return Value
String

```
alert(ju.dragdrop.ver);

// It will show 1.0.0

if(ju.dragdrop.ver.equalTo('1.0.0')){
        /* rest of the code goes here */
}
```

Example 7.1

---

enable

This method will make HTML Element(s) 'draggable' in the page. With help of this method you can make single HTML element (object) as drag gable or multiple elements like list sorting, dragging between two groups or more groups.

Features:

- Make any HTML element (Object) drag gable in the web page
- Customized ordering within the list
- Customized ordering within and between groups
- Easy to retrieve original order of values
- Easy to retrieve newly ordered values

Parameters

*HTML-Element-ID	
onDrag-start-action	- function -> action to be performed on drag starts
onDrag-action	- function -> action to be performed on dragging
onDrag-end-action	- function -> action to be performed on drag completes

* Mandatory

Return Value
Object

```
ju.dragdrop.enable("flower");

//for flower element id dragging will be enabled
```

Example 7.2

---

disable

This API disables the drag gable mode of HTML element on the fly in web page.

Parameters
HTML-Element-ID or Object

Return Value

True	-> Success
False	-> Failed or Error Occurred
Null	-> For this object or HTML Element dragging is not enabled

```
//Explicit Usage

ju.dragdrop.disable("flower");

//Implicit Usage

flower.disable( );              //flower is the object; already dragging enabled
```

Example 7.3

---

getOriginalValues[1]

Returns initial values of the list or group.

Parameters - None

Return Value
Array

```
group1.getOriginalValues( );

//returns “” in an Array format
```

Example 7.4

---

getValues[1]

Returns current (manipulated or selected) values of the list or group.

Parameters - None

Return Value
Array

```
group1.getValues( );

//returns “” in an Array format
```

Example 7.5

---

reset[1]

Restores initial or at the time of load values to the list or Object.

Parameters - None

Return Value
Array

```
group1.reset( );
```

Example 7.6

[1] This API is applicable to list or group object.

Chapter

8

8 juDateTime v1.1

---

ver

Returns version of juDateTime[13] plug-in being use.

Parameters - None

Return Value
String

```
alert(ju.datetime.ver);

// It will show 1.0.0

if(ju.datetime.ver.equalTo('1.0.0')){
        /* rest of the code goes here */
}
```

Example 8.1

---

getDate[1]

Return the System Date with specified Mask[1].

Parameters
Mask - default mask is "shortDate"

Return Value
dateString

```
ju.datetime.getDate();        //Default masked value will be return.
//Eg: 6/9/07

ju.datetime.getDate("longDate");        //Specified masked value will be return.
//Eg: June 9, 2007
```

Example 8.2

[13] juDateTime implemented as per ISO 8601. http://en.wikipedia.org/wiki/ISO_8601

showDate[1]

Display the System Date in the given HTML Element ID with specified Mask[1].

Parameters
HTML-Element-Id
Mask - default mask value is “shortDate”

Return Value
null - if given element id invalid or parameter missing.

```
ju.datetime.showDate();        //Default masked value will be return.
//Eg: 6/9/07

ju.datetime.showDate(“longDate”);        //Specified masked value will be return.
//Eg: June 9, 2007
```

Example 8.3

getTime[2]

Return the System time with specified Mask[2].

Parameters
Mask - default mask value is “shortTime”

Return Value
timeString

```
ju.datetime.getTime();        //Default masked value will be return.
//Eg: 5:46 PM

ju.datetime.getTime(“longTime”);        //Specified masked value will be return.
//Eg: 5:46:21 PM EST
```

Example 8.4

showTime[2]

Display the System Time in the given HTML Element ID with specified Mask[2].

Parameters
HTML-Element-Id
Mask - default mask value is “shortTime”

Return Value
null - if given element id invalid or parameter missing.

```
ju.datetime.showTime();         //Default masked value will be shown.
//Eg: 5:46 PM

ju.datetime.showTime(“longTime”);         //Specified masked value will be shown.
//Eg: 5:46:21 PM EST
```

Example 8.5

getDateTime[3]

Return the System DateTime with specified Mask[3].

Parameters
Mask - default mask value is “isoDateTime”

Return Value
dateTimeString

```
ju.datetime.getDateTime();         //ISO DateTime value will be return.
//Eg: 2008-09-09T17:46:21

ju.datetime.getDateTime(“isoUtcDateTime”); //ISO UTC DateTime will be return.
//Eg: 2008-09-09T22:46:21Z
```

Example 8.6

[1] Following Masking is applies to getDate, showDate API

Parameter	Mask	Example
shortDate	m/d/yy	6/9/07
mediumDate	mmm d, yyyy	Jun 9, 2007
longDate	mmmm d, yyyy	June 9, 2007
fullDate	dddd, mmmm d, yyyy	Saturday, June 9, 2007
isoDate	yyyy-mm-dd	2007-06-09

[2] Following Masking is applies to getTime, showTime API

Parameter	Mask	Example
shortTime	h:MM TT	5:46 PM
mediumTime	h:MM:ss TT	5:46:21 PM
longTime	h:MM:ss TT Z	5:46:21 PM EST
isoTime	HH:MM:ss	17:46:21

[3] Following Masking is applies to getDateTime API

Parameter	Mask	Example
isoDateTime	yyyy-mm-dd'T'HH:MM:ss	2007-06-09T17:46:21
isoUtcDateTime	UTC:yyyy-mm-dd'T'HH:MM:ss'Z'	2007-06-09T22:46:21Z