

v1.6.0	JS365Lib2.js では関数呼び出し時に名前空間 JS365Lib を使う必要がある。				
No	関数	パラメータ	戻り値	機能	備考
1	escURL(str)	str: 文字列	URL 特殊文字をエスケープした文字列	URL 特殊文字をエスケープした文字列を返す。	
2	escHTML(str)	str: 文字列	HTML 特殊文字をエスケープした文字列	HTML 特殊文字をエスケープした文字列を返す。	
3	E(id, type="i", index=0)	id: HTML要素の id, name, class, tag type: パラメータ id の種類 index: typeがid以外の時の場合のインデックス	パラメータで指定された HTML 要素 (オブジェクト)	HTML 要素 (オブジェクト) を返す。 type="i" (デフォルト) は id, type="n" は name type="c" は class type="t" は tag	
4	getValue(id)	id: 要素の id または要素 (オブジェクト)	要素の値または内部文字列	要素の値を得る。	p タグ等の文字列を含む。
5	setValue(id, value, escape=true)	id: 要素の id または要素 (オブジェクト) value: 値 escape=true: HTML特殊文字をエスケープするか どうか	開始タグの直後	要素の値を設定する。	p タグ等の文字列を含む。
6	insertHTML(id, html, position=0)	id: 要素の id または要素 (オブジェクト) html: 挿入するHTML position: 0: 開始タグの直前 1: 開始タグの直後 2: 終了タグの直前 3: 終了タグの直後	開始タグの直後	HTML文字列をタグの前後に挿入する。	
7	create(tag, elem=null)	tag: タグ名 elem: null でない場合、この要素の後に挿入する。	タグ要素 (オブジェクト)	要素を作成する。 elem が null でない場合は、その要素の後に作成した要素をに挿入する。	
8	addChild(parent, tag)	parent: 親要素オブジェクトまたは id tag: 子要素のタグ名	子要素	子要素を作成する。	
9	getAttr(id, attr)	id: 要素の id または要素 (オブジェクト) attr: 属性名	属性の値	要素の属性を得る。	
10	setAttr(id, attr, value)	id: 要素の id または要素 (オブジェクト) attr: 属性名 value: 属性の値	なし	要素の属性を設定する。	
11	dropAttr(id, attr)	id: 要素の id または要素 (オブジェクト) attr: 属性名	なし	要素の属性を削除する。	
12	getText(url, callback)	url :サーバのURL callback: コールバックするパラメータが1個の関数。	なし	指定した完全なリクエストパス (URL) から GET メソッドでテキストを得る。	url はパラメータを含む。 callback(str)
13	getJSON(url, callback)	url :サーバのURL callback: コールバックするパラメータが1個の関数。	なし	指定した完全なリクエストパス (URL) から GET メソッドで JSON を得る。	url はパラメータを含む。 callback(data)
14	fetchText(url, data, method, callback)	url: パラメータを含まないサーバのURI data: パラメータ(JSON) method: GET または POST callback: コールバックするパラメータが1個の関数	コールバック関数のパラメータは文字列 (plain/text)	指定した URL から GET/POST メソッドで JSON を得る。	GET,POST 以外はサポートしない。 callback(str)
15	fetchJSON(url, data, method, callback)	url: パラメータを含まないサーバのURI data: パラメータ(JSON) method: GET または POST callback: コールバックするパラメータが1個の関数	コールバック関数のパラメータはJSON (application/json)	指定した URL から GET/POST メソッドで BLOB (画像など) を得る。	GET,POST 以外はサポートしない。 callback(data)
16	fetchBLOB(url, data, method, as, callback)	url: パラメータを含まないサーバのURI data: パラメータ(JSON) method: GET または POST as: BLOB をそのまま返すか、文字列などにして返すかを指定する。 callback: コールバックするパラメータが1個の関数	コールバック関数のパラメータは BLOB。 JPEG画像なら image/jpeg	指定した URL から GET/POST メソッドで BLOB を得る。 ただし、as="text" なら文字列、as="arrayBuffer" なら ArrayBuffer 、"slice" なら Slice、"stream" なら Stream にして返す。as="" がデフォルト。	GET,POST 以外はサポートしない。 callback(blob)

17	fetchArrayBuffer(url, data, method, dataView, callback)	url: パラメータを含まないサーバのURI data: パラメータ(JSON) method: GET または POST callback: コールバックするパラメータが1個の関数	コールバック関数のパラメータはバイト列など	指定した URL から GET/POST メソッドで ArrayBuffer を得る。ただし、dataView=true の場合は DataView にしてから返す。	callback(buffer)
18	postForm(url, form, callback, event=null)	url: パラメータを含まないサーバのURI form: フォーム (オブジェクトまたは id)。 callback: コールバックするパラメータが1個の関数 event: イベントリスナが受け取った Event オブジェクト	コールバック関数へ戻り値が返される。	フォームの内容をFormDataオブジェクトに変換してサーバへポストする。	input[type="file"] を含む enctype="multipart/form-data" 指定のフォームも可能 event が null でないときは、preventDefault() を実行する。 callback(data)
19	postFormData(url, formData, callback, event=null)	url: パラメータを含まないサーバのURI formData: FormDataオブジェクト callback: コールバックするパラメータが1個の関数 event: イベントリスナが受け取った Event オブジェクト	コールバック関数へ戻り値(JSON)が返される。	FormDataオブジェクトをサーバへポストする。	event が null でないときは、preventDefault() を実行する。 callback(data)
20	postJSON(url, data, callback, event=null)	url: パラメータを含まないサーバのURI data: JSON (オブジェクト)。 callback: コールバックするパラメータが1個の関数 event: イベントリスナが受け取った Event オブジェクト	コールバック関数へ戻り値(JSON)が返される。	formData をサーバへPOSTする。	event が null でないときは、preventDefault() を実行する。 callback(data)
21	clickEvent(id, callback)	id: イベントを発生させる要素の id または要素オブジェクト callback: イベントハンドラ	なし	click イベントハンドラを追加する。	ボタン (button, input[type="button"])などで使う。 callback()
22	changeEvent(id, callback)	id: イベントを発生させる要素の id または要素オブジェクト callback: イベントハンドラ	なし	change イベントハンドラを追加する。	ドロップダウン (select) などを使う。 callback()
23	uploadTextFile(url, id, callback, event=null)	url: サーバの URL id: input[type=file] の id callback: アップロードが完了したときコールされる。 event: イベントリスナが受け取った Event オブジェクト	なし	input[type="file"] コントロールで指定されるテキストファイルをアップロードする。	event が null でないときは、preventDefault() を実行する。 callback(text)
24	uploadBinaryFile(url, id, callback, event=null)	url: サーバの URL id: input[type=file] の id callback: アップロードが完了したときコールされる。 event: イベントリスナが受け取った Event オブジェクト	なし	input[type="file"] コントロールで指定されるバイナリーファイルをArrayBufferとしてアップロードする。	event が null でないときは、preventDefault() を実行する。 callback(text)
25	uploadBLOB(url, blob, callback, event=null)	url: サーバの URL blob: Blobオブジェクト callback: アップロードが完了したときコールされる。 event: イベントリスナが受け取った Event オブジェクト	なし	Blob オブジェクトをアップロードする。	event が null でないときは、preventDefault() を実行する。 callback(text)
26	readTextFile(id, callback, event=null)	id: input[type=file] の id callback: アップロードが完了したときコールされる。 event: イベントリスナが受け取った Event オブジェクト	なし	File API を使ってテキストファイルを読む。	event が null でないときは、preventDefault() を実行する。 callback(text)
27	readBinaryFile(id, callback, event=null)	id: input[type=file] の id callback: アップロードが完了したときコールされる。 event: イベントリスナが受け取った Event オブジェクト	なし	File API を使ってバイナリーファイルを読む。	event が null でないときは、preventDefault() を実行する。 callback(arrayBuffer)
28	getStorageKeys(prefix="", session=true)	prefix="": キーの前に付ける文字列。 session: true なら sessionStorage、false なら localStorage	キーの配列	ストレージのキーの一覧を得る。 prefix が空でないときはその文字列が先頭にあるキー (prefix を除いたもの) だけを取得する。	localStorage は他のアプリケーションと共有するため、prefix を付けないとキーが競合する

29	getStorageValue(key, prefix="", session=true)	key: 取得する値に対するキー。 prefix: キーの前に付ける文字列。 session: true なら sessionStorage、false なら localStorage	ストレージのキーに対する値	キーに対する値を得る。	localStorage は他のアプリケーションと共有するため、prefix を付けないとキーが競合する
30	setStorageValue(key, value, prefix="", session=true)	key: キー value: 値 prefix: キーの前に付ける文字列。 session: true なら sessionStorage、false なら localStorage	なし	キーに対する値を追加する。ただし、キーが既存の場合は上書きする。	localStorage は他のアプリケーションと共有するため、prefix を付けないとキーが競合する
31	clearSessionStorage()	なし	なし	sessionStorage の内容を削除する。	localStorage は他のアプリケーションと共有するため個別のキーを削除することによりクリアすること
32	deleteStorageKey(key, prefix="", session=true)	key: 削除するキー prefix: キーの前に付ける文字列。 session: true なら sessionStorage、false なら localStorage	なし	キーに対する項目を削除する。	localStorage は他のアプリケーションと共有するため、prefix を付けないとキーが競合する
33	htmlTable(rows, header=false, table="", tr="", th="", td="")	rows: テーブル行のデータ。カラムデータの配列である。 header: true なら rows の先頭行を表題とみなす。 table: table タグのクラス tr: tr タグのクラス th: th タグのクラス td: td タグのクラス	HTML	HTML テーブルを作成する。	
34	htmlList(data, type="ul", ul="", li="")	data: リストのデータで配列型。 type: "ul" か "ol" ul: ul または ol タグのクラス。 li: li タグのクラス	HTML	HTML リストを作成する。	
35	onDragEnter(event)	event: DragEnter イベント	なし	onDragEnter ハンドラ。ドロップ領域の onDragEnter イベントハンドラに登録する。	
36	onDragOver(event)	event: DragOver イベント	なし	onDragOver ハンドラ。ドロップ領域の onDragOver イベントハンドラに登録する。	
37	onDrop(event, control, listid)	event: Drop イベント control: input[type="file"] コントロール listid: ul タグの id	なし	onDrop ハンドラ。ドロップ領域の onDrop イベントハンドラに登録する。	control はオブジェクト。 listid は ul タグの id で、そこにドラッグされたファイルを表示する。