Search...

Getting started
Components
Utilities
- Border
- Clear float
- Colors
- Cursors
- Display & visibility
- Embed
- Extensions
- Flex
- **Functions**
- Float
- Position
- Sizing
- Spacing

Animations
Additional

Functions

Metro 4 implements several useful functions that are accessible through the special object.

About

All special utilities functions stored in `Metro.utils` object.

Function	Desc
`isUrl(v)`	Checks if value is url
`isTag(v)`	Checks if value is tag.
`isColor(v)`	Checks if value is hex color. #fff, #ffffff - true
`isEmbedObject(v)`	Checks if value is embed object, value to check - html string
`isVideoUrl(v)`	Checks if value is youtube or vimeo video link
`isDate(v)`	Checks if value is valid date string
`isInt(v)`	Checks if value is integer
`isFloat(v)`	Checks if value is float
`isTouchDevice()`	Checks if device is touchable
`isFunc(v)`	Checks if value is function
`isObject(v)`	Checks if value is object
`isArray(v)`	Checks if value is array
`isType(v, t)`	Checks if value is a type: isType(v, 'function')
`isMetroObject(el, type)`	Checks if element is a Metro component
`isJQuery(el)`	Checks if element is a jQuery object
`isM4Q(el)`	Checks if element is a m4q object
`isQ(el)`	Checks if element is a m4q or jQuery object
`embedObject(el)`	Insert element into embed container. Return html string
`embedUrl(el)`	Insert youtube link into embed container. Return html string
`secondsToTime(seconds)`	Return object {hours, minutes, seconds}
`hex2rgba(hex, alpha)`	Return string 'rgba(x, x, x, a)'
`random(from, to)`	Return random value
`uniqueId()`	Return string in format xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx
`elementId(prefix)`	Return string. Based in datetime
`secondsToFormattedString(seconds)`	Return string "hh:mm:ss"
`callback(f, args, context)`	Synonym to exec function
`exec(f, args, context)`	Exec function or code with arguments and context
`isOutsider(el)`	Check element to outside of the view
`inViewport(el)`	Check element in viewport
`objectLength(el)`	Return object length.
`percent(a, b, r)`	Return percent value for b from a. If r is true, returned value is integer else - real
`camelCase(str)`	Return 'data-on-value' as 'dataOnValue'
`objectShift(obj)`	Return shifted object. Remove element with first key
`objectDelete(obj, key)`	Delete key from object and return it
`objectClone(obj)`	Copy object to clone and return it
`arrayDelete(arr, val)`	Delete from array by value and return it
`arrayDeleteByKey(arr, key)`	Delete key from array and return it
`arrayDeleteByMultipleKeys(arr, keys_array)`	Delete key from array and return new array
`nvl(data, other)`	Check data for null value and return other if null
`github(repo, callback)`	Get information from github and put then in callback
`detectIE()`	Detect if browser is Internet explorer or Edge
`detectChrome()`	Detect if browser is Chrome
`md5(str)`	Encode string with md5 algorithm

Table of contents

Functions

Function	Desc
`encodeUri(str)`	Fix to base eencodeUri function
`pageHeight()`	Return real page height
`cleanPreCode(selector)`	Remove white spaces from code
`coords(el)`	Return element coordinates as object {top, left}
`positionXY(event, type)`	Return position for type: screen, page, client
`clientXY(event)`	Return position
`pageXY(event)`	Return position
`screenXY(event)`	Return position
`isRightMouse(event)`	Check if user click on right mouse button
`hiddenElementSize(el, includeMargin)`	Return element size as object {width, height}
`getStyle(el, pseudo)`	Return element calculated styles
`getStyleOne(el, property)`	Return element calculated styles for specified property
`getTransformMatrix(el)`	Return element transform matrix
`computedRgbToHex(rgb)`	Return hex value for computed element color.
`computedRgbToHex(rgb)`	Return hex value for rgb string 'rgb(x, x, x)' => #xxxxxx
`computedRgbToRgba(rgb, alpha)`	Return rgba string for rgb string 'rgb(x, x, x)' => 'rgba(x, x, x, a)'
`computedRgbToArray(rgb)`	Return array for string 'rgb(x, x, x)' => [x, x, x]
`hexColorToArray(c)`	Convert hex color value to array
`hexColorToRgbA(c, a)`	Convert hex color value rgba string: '#xxxxxx' => 'rgba(x, x, x, a)'
`getInlineStyles(el)`	Return element inline styles as array
`updateURIParameter(uri, key, val)`	Update parameter in Uri
`getURIParameter(uri, key)`	get parameter from Uri
`getLocales()`	Get registered metro locales
`addLocale(data)`	Register metro locale in runtime
`strToArray(str, delimiter)`	Convert string to array
`aspectRatioH(width, ratio)`	Return height for specific ratio
`aspectRatioW(height, ratio)`	Return width for specific ratio
`valueInObject(obj, value)`	Check if value exist in object
`keyInObject(obj, key)`	Check if key exist in object
`newCssSheet(media)`	Create css sheet object
`addCssRule(sheet, selector, rules, index)`	Add rule to css sheet object
`media(query)`	Check media query
`mediaModes()`	Get current media points
`mediaExist(m)`	Return true if point exist in current medias
`inMedia(m)`	Check if point is current media
`isValue(val)`	Return true if val not in [undefined, null, ""]
`isNegative(val)`	Return true if val less then 0
`isPositive(val)`	Return true if val more then 0
`isZero(val)`	Return true if val is 0 (int or float)
`between(val, bottom, top, equals)`	Return true if val between bottom and top
`parseMoney(val)`	Return numeric value from any money format. Ex: $5,640.63 -> 5640.63
`func(string)`	Return function object
`nearest(val, precision, down)`	search for the nearest integer, a multiple of required. Ex: Metro.utils.nearest(37, 5, false) -> 40, Metro.utils.nearest(37, 5, true) -> 35
`copy(el)`	Copy element to clipboard
`isLocalhost()`	Check if current location is localhost
`getCursorPosition(element, event)`	Return mouse or pointer position as x, y
`getCursorPositionX(element, event)`	Return mouse or pointer position x
`getCursorPositionY(element, event)`	Return mouse or pointer position y
`formData(form)`	Return named inputs as obj as pairs `name: value`

Function	Desc