

Tint.Doc

*=O= means optional
remember false==0; true==1

→ Functions

Find(a,b,c) returns an array of nodes/elements *a* or *b* each element's attribute(s) or array position(s). no arguments == '*' == all nodes

a =O= 1st: document.querySelectorAll(**a**), **a** == css style selector [Selectors](#), String or Arrays of Strings or Arrays, returns same form 2nd: window[**a**] or path 3rd: itself

b =O= return each value *a*=*a*[*b*], accepts string or array of strings or arrays, returns same form

c =O= root element for Find(), default == document

ex: Find('button','offsetWidth') ; Find('*',['id','name','offsetLeft']) ; Find('*', 'value', Find('form')[3])

lambda(a,b,c) returns a copy with mirrored values of the data for manipulation

a == a function being ran over each value of the data, three variables for Value, Key, Data

b == the data (object, array, string, any type)

c =O= set data type **new c()**

ex: lambda(function(x){return x}, 'this into array', Array), lambda(function(x){return x}, Find('button')[0]),

range(a,b,c) creates an array of numbers based on the inputs.

a == only *a*: array of numerical values 0 through (*a*-1), incremented by 1 for each value

b =O= if only *a* and *b*: array of numerical values *a*..*b* incremented by 1 for each value

c =O= array of numerical values *a* *through* (=<*b*) incremented by **c**

ex: range(5) == [0,1,2,3,4] ; range(6,9) == [6,7,8,9] ; range(3,15,4) == [3,7,11,15]

GetFile(a,b,c) returns the contents of any file, with a parent *b*: the data is loaded and transferred to the parent.

a == the address of the file being loaded or an array of addresses||arrays

b =O= the element selected to be the parent of *a*, will load before transferring

c =O= style selector for which nodes are transferred

ex: GetFile('results.txt') ; GetFile('somePhoto.jpg' , Find('#ans')[0]) ;
GetFile('template.html', leftMenu, '.left>button')

Data(a,b,c) controls retrieving/saving temporary/permanent data on a browser.

a == Array or String: name(s) of the data to retrieve; Object: key/value(type==String) pairs for setting data

b =O= true if the data is temporary (sessionStorage)

c =O= true if you wish to delete the data permanently.

ex: Data('names') ; Data({dates:[1985,1948,1962,1957].join('-')},false) ; Data('user',false,true)

Make(a,b,c,d,e) makes node *a*, sets attributes *b*, sets events *c*, adds innerHTML *d*, and appends it to *e*.

a == a string or array of strings representing the node(s) tag name(s)

b =O= an object with key/value pairs for assigning attributes

c =O= an object with key/value pairs for assigning events

d =O= the innerHTML for the node

e =O= the node the new node will be appended to

ex: Make('button') ; Make('input' ,{type:'button',value:Find('textarea')[2].value},{}, " , Find('form')[3])

TTS(a) simplifies javascript's SpeechSynthesis api

a == if type is String: reads the string, if Object: sets properties and reads the value "text"

ex: TTS("Hello World!") ; TTS({text:"What was that?",rate:2,volume:0.5})

SpeechRec(a,b) simplifies javascript's SpeechRecognition api

a == a function with 1-2 arguments, first: the final transcript, second: the event Object

b =O= Object with key/value pairs setting properties of the SpeechRecognition Object

ex: SpeechRec(function(**TS**){inputNode.value = **TS**},{“onspeechend”:function(e){e.target.stop()}}).start() ;

sequence(a) performs a sequence of timed functions, returns an array for **cancel()**

a == an array of arrays containing setTimeout(code , seconds) pairs

ex:

sequence([

['alert(3)',3],

['alert(8)',8],

['alert(12)',12]]) ;

sequence(range(0,5,60).each(function(x){return [function(){alert(x+' seconds')},x]}))

cancel(a) to stop animations and sequences.

a == an array returned by .animate() or sequence()

ex: cancel(Find('button').animate(...)) ; cancel(sequence(...))

protect(a) protects pictures from easy user download.

a =O= optional Array of nodes to protect, default: Find('img')

ex: protect(Find('img[protected]')) ; protect(Find('.artwork'))

ord(a) changes number *a* to ascii character.

a == a number representing a character

ex: ord(70)=="F" ; ord(117)=="u" ; ord(99)=="c" ; ord(110)=="n" ; ord(65)=="A"

http://www.w3schools.com/tags/ref_ascii.asp

http://www.w3schools.com/tags/ref_entities.asp

http://www.w3schools.com/tags/ref_symbols.asp

hex(a) converts a hex value characters (between 0-9||a-f) to a ascii character.

a == a 2 character hex code for a character value

ex: hex("50")=="P" ; hex("3e")==">" ; hex("58")=="X"

http://www.w3schools.com/tags/ref_urlencode.asp

bin(a) converts a binary code string into an ascii character

a == the binary code

ex: bin("00001001")=="9" ; bin("00001011")=="11"

→ Array Prototypes

.cycle(a,b) cycles through an Array of codes with each activation

a == an Array of functions ordered to evaluate 1 each time activated, the function has 1 to 2 inputs for the item and the key.

b =O= a String containing the key each array value uses to store the cycle position

ex:

Find('.toggle').cycle([function(x){x.firstChild.style.display='none'},function(x){x.firstChild.style.display='block'}], 'hideFirst') ; Find('p').cycle([function(x,k){return x.parent},function(x,k){return x.innerHTML}])

.set(a,b,c,d) set the attributes and events of a list of nodes

a =O= an object with key/value pairs to set attributes of the nodes

b =O= an objects with key/value pairs to set events of the nodes

c =O= any html to be injected

d =O= a parent element to be attached to

ex: Find('button').set({name:'test'}, {onclick:function(e){alert(e.data)}}, "!", Find("div")[5])

.animate(a,b) sequence of timed style changes to an Array of nodes/objects.

a == a string of style keys separated by a comma ' , ' mapped to b[0] values

b == an array of arrays with 2 items, an array b[0] of values and a number b[1] seconds before style change

ex: Find('.blink').animate('background,color',[[['white','black'],0],[['black','white'],1]])

.items(a) used to gather a general/specific list of items in the Array

a == a String or Array of strings containing a number (even negative) for the item index or numbers separated by colons as range() input Array style indices.

ex: cars.items(['9','45','86']); users.items('27:122:3'); links.items(['50:80',['4','6:20','99']])

.copy(a) used to make a copy of an array for unrelated editing

a =O= number of copies

ex: [1,2,3].copy(3) == [1,2,3,1,2,3,1,2,3]; userData.copy()

.random(a,b,c) with no arguments returns a random value from the array.

a =O= amount of random values in the array to return

b =O= true if you want no repeat

c =O= true if you want to remove the random selections from the original Array

ex: Find().random(); range(5000).random(5000,1); dice=range(6).random(2)

.removeNodes(a) removes the nodes in the array.

a =O= true if you want to leave the innerHTML in it's place

ex: Find('img').removeNodes(); Find('div?offsetLeft>500').removeNodes(1)

.changeTag(a) changes the tags of the element array

a == a new tag for each element

ex: Find('button').changeTag('input'); Find('iframe').changeTag('img')

.each(a) function ran over each element

a == function with 3 values for the Value, Index, and Array

ex: Find('button').each(function(v,i,a){return v.parentNode})

.eval(a,b) evaluate a function over a whole Array

a == function with 1 or 2 inputs, the Array and argument b

b =O= an argument to be passed into the eval function

ex: [2,3,4,7,8].eval(function(a,b){return b>a.random()},5)

.sortBy(a,b) sorts an array according to a value specified by this[a], no arguments sorts regularly. String values automatically sorted alphabetically and Number values numerically.

a =O= an index, string representing a key, or function returning the value to sort by

b =O= function to give the .sort(b) prototype

ex: [[bill,2],[scott,4],[sara,1],[jill,3]].sortBy(1); Find('button').sortBy('offsetTop');
Find('input').sortBy('value',function(a,b){return b-a}); Find('img').sortBy(function(x){return x.offsetHeight*x.offsetWidth})

.zip(a,...) makes a new array with arrays containing values of equal index

a == the 2nd array in the zip, more arrays can be added as arguments

ex: [1,2,3].zip([9,8,7]) == [[1,9],[2,8],[3,7]] ; [1,2].zip([3,4],[5,6]) == [[1,3,5],[2,4,6]]

.transpose() returns a copy of the array with all arrays zipped

ex: [[1,2,3],[4,5,6]].transpose() == [[1,4],[2,5],[3,6]] ; Find(['.title','.add','.logo']).transpose()

.perm(a,b) mixes an array into an array of unique combinations of **a** sized groups

a = 0 = length of the unique combinations

b = 0 = each combination .join(**b**)

ex: [0,1].perm(2) == [[0,0],[0,1],[1,0],[1,1]] ; [5,7].perm(2,"") == ['55','57','75','77']

.rotate(a) returns a new rotated version of the array

a = 0 = a positive or negative number determining the direction and degree of rotation

ex: [1,2,3,4].rotate() == [2,3,4,1] ; [1,2,3,4,5,6].rotate(-2) == [5,6,1,2,3,4]

.swap(a,b) used to swap values in an array.

a == the index of the first value to be swapped.

b == the index of the second value to be swapped.

ex: [1,2,3,4,5].swap(2,4) == [1,2,5,4,3] ;

// Removed in newer versions, redundant

.all(a) used to check if all items or a key of each item all are true based on **a** (returns true if all equal the value, false otherwise)

a == a function with 1 variable representing each item, returns a truth assertion

ex: [5,5,5].all(function(x){return x == 5}) == true ; [6,4,2,3].all(function(x){return x>3}) == false ;

Find().all(function(x){x.parentNode == body}) == false

// Removed in newer versions, redundant

.any(a) same as .all() but returns true if 1 item tests as true, false if none are true

.check(a) makes a console.log of the data and returns the data

a = 0 = extra data to be shown in the log

ex: Find('input').check().random() ; [false,true].perm(8).check()

→ Number Prototypes

.random(a,b) selects a random whole number between 0 and the original number

a = 0 = minimum possible answer [a..this] (inclusive), if it's more than the original number it will select a number between but excluding both (this..a) (exclusive).

b = 0 = function to be used on the answer (default = Math.floor)

ex: (7).random() == (0..7) ; (80).random(50) == (50..80) ; (30).random(90) == (31..89)

→ String Prototypes

.to(a) converts a string into a hex,ord, or a default map() function to an Array()

a == functions ord, bin, and hex can be used to specify what to convert the characters to. Any function can be used in a map function use.

ex: 'this will change to an array of numbers 0-255'.to(ord) ; 'this will be an array of 2 #[a-f]'.to(hex)

→ Object Prototypes

→ Objects

Params[a] returns the Parameters of the address (values after ?)

a == name of variable in parameter

ex: fakesite.com?num=67667&page=home , Params['num']== '67667' , Params.page== 'home'
