Array

Aug 23, 2020 20:33

Supported Clients

SmartClient WebClient NGClient MobileClient

Property Summary

Object	[index]	Get an element by index.
Number	length	Get the length of the array.

Methods Summary

Array	concat(value1)	Returns a new array comprised of this array joined with other array(s) and/or value(s).
Array	concat(value1, value2)	Returns a new array comprised of this array joined with other array(s) and/or value(s).
Array	concat(value1, value2, valueN)	Returns a new array comprised of this array joined with other array(s) and/or value(s).
Boolean	every(callback)	Runs a function on items in the array while that function is returning true.
Boolean	every(callback, thisObject)	Runs a function on items in the array while that function is returning true.
Array	filter(callback)	Runs a function on every item in the array and returns an array of all items for which the function returns true.
Array	filter(callback, thisObject)	Runs a function on every item in the array and returns an array of all items for which the function returns true.
void	forEach(callback)	Runs a function (callback) on every item in the array.
void	forEach(callback, thisObject)	Runs a function (callback) on every item in the array.
Number	indexOf(searchElement)	Returns the first index at which a given element can be found in the array, or -1 if it is not present.
Number	indexOf(searchElement, fromIndex)	Returns the first index at which a given element can be found in the array, or -1 if it is not present.
Boolean	isArray(obj)	Checks whether an object is an array or not.
String	join(delimiter)	Puts all elements in the array into a string, separating each element with the specified delimiter
Number	lastIndexOf(searchElement)	Returns the last index at which a given element can be found in the array, or -1 if it is not present.
Number	lastIndexOf(searchElement, fromIndex)	Returns the last index at which a given element can be found in the array, or -1 if it is not present.
Array	map(callback)	Runs a function on every item in the array and returns the results in an array.
Array	map(callback, thisObject)	Runs a function on every item in the array and returns the results in an array.
Object	pop()	Pops the last string off the array and returns it.
Number	push(value1)	Mutates an array by appending the given elements and returning the new length of the array.
Number	push(value1, value2)	Mutates an array by appending the given elements and returning the new length of the array.
Number	push(value1, value2, valueN)	Mutates an array by appending the given elements and returning the new length of the array.
Object	reduce(f, initialValue)	Reduces the array to a single value by executing a provided function for each value of the array (from left-to-right).
Array	reverse()	Puts array elements in reverse order.
Object	shift()	Decreases array element size by one by shifting the first element off the array and returning it.
Array	slice(begin)	The slice method creates a new array from a selected section of an array.
Array	slice(begin, end)	The slice method creates a new array from a selected section of an array.
Boolean	some(callback)	Runs a function on items in the array while that function returns false.
Boolean	some(callback, thisObject)	Runs a function on items in the array while that function returns false.
Array	sort()	Sorts the array elements in dictionary order or using a compare function passed to the method.
Array	sort(function)	Sorts the array elements in dictionary order or using a compare function passed to the method.
Array	splice(arrayIndex, length)	It is used to take elements out of an array and replace them with those specified.
Array	splice(arrayIndex, length, value1)	It is used to take elements out of an array and replace them with those specified.
Array	splice(arrayIndex, length, value1, value2)	It is used to take elements out of an array and replace them with those specified.
Array	splice(arrayIndex, length, value1, value2, valueN)	It is used to take elements out of an array and replace them with those specified.
Number	unshift(value1, value2, valueN)	Places element data at the start of an array.

Property Details

[index]

```
Get an element by index.
```

Returns

Object

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
array[0]
```

length

Get the length of the array.

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
array.length
```

Methods Details

concat(value1)

Returns a new array comprised of this array joined with other array(s) and/or value(s).

Parameters

Object value1 ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
array.concat();
```

concat(value1, value2)

Returns a new array comprised of this array joined with other array(s) and/or value(s).

Parameters

Object value1 ;
Object value2 ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
array.concat();
```

concat(value1, value2, valueN)

Returns a new array comprised of this array joined with other array(s) and/or value(s).

Parameters

Object value1 ;
Object value2 ;
Object valueN ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
array.concat();
```

every(callback)

Runs a function on items in the array while that function is returning true. It returns true if the function returns true for every item it could visit.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function isNumber(value) { return typeof value == 'number'; }
var a1 = [1, 2, 3];
application.output(a1.every(isNumber));
var a2 = [1, '2', 3];
application.output(a2.every(isNumber));
```

every(callback, thisObject)

Runs a function on items in the array while that function is returning true. It returns true if the function returns true for every item it could visit.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;
Array thisObject ;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function isNumber(value) { return typeof value == 'number'; }
var a1 = [1, 2, 3];
application.output(a1.every(isNumber));
var a2 = [1, '2', 3];
application.output(a2.every(isNumber));
```

filter(callback)

Runs a function on every item in the array and returns an array of all items for which the function returns true.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a1 = ['a', 10, 'b', 20, 'c', 30];
var a2 = a1.filter(function(item) { return typeof item == 'number'; });
application.output(a2);
```

filter(callback, thisObject)

Runs a function on every item in the array and returns an array of all items for which the function returns true.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;
Array thisObject ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a1 = ['a', 10, 'b', 20, 'c', 30];
var a2 = a1.filter(function(item) { return typeof item == 'number'; });
application.output(a2);
```

forEach(callback)

Runs a function (callback) on every item in the array. The callback function is invoked only for indexes of the array which have assigned values.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function printThemOut(element, index, array) {
                application.output("a[" + index + "] = " + element);
}
var a = ['a', 'b', 'c'];
a.forEach(printThemOut);
```

forEach(callback, thisObject)

Runs a function (callback) on every item in the array. The callback function is invoked only for indexes of the array which have assigned values.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;
Object thisObject ;

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function printThemOut(element, index, array) {
                application.output("a[" + index + "] = " + element);
}
var a = ['a', 'b', 'c'];
a.forEach(printThemOut);
```

indexOf(searchElement)

Returns the first index at which a given element can be found in the array, or -1 if it is not present.

Parameters

Object searchElement ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = ['a', 'b', 'a', 'b', 'a'];
application.output(a.indexOf('b'));
application.output(a.indexOf('b', 2));
application.output(a.indexOf('z'));
```

indexOf(searchElement, fromIndex)

Returns the first index at which a given element can be found in the array, or -1 if it is not present.

Parameters

Object searchElement ;
Number fromIndex ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = ['a', 'b', 'a', 'b', 'a'];
application.output(a.indexOf('b'));
application.output(a.indexOf('b', 2));
application.output(a.indexOf('z'));
```

isArray(obj)

Checks whether an object is an array or not.

Parameters

Object obj ;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = [1, 2, 3];
application.output(Array.isArray(a)); //prints true
application.output(Array.isArray(23)); //prints false
```

join(delimiter)

Puts all elements in the array into a string, separating each element with the specified delimiter

Parameters

String delimiter ;

Returns

String

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var jwords = words.join(";");
```

lastIndexOf(searchElement)

Returns the last index at which a given element can be found in the array, or -1 if it is not present. The array is searched backwards, starting at fromIndex.

Parameters

Object searchElement ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = ['a', 'b', 'c', 'd', 'a', 'b'];
application.output(a.lastIndexOf('b'));
application.output(a.lastIndexOf('b', 4));
application.output(a.lastIndexOf('z'));
```

lastIndexOf(searchElement, fromIndex)

Returns the last index at which a given element can be found in the array, or -1 if it is not present. The array is searched backwards, starting at fromIndex.

Parameters

Object searchElement ;
Number fromIndex ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = ['a', 'b', 'c', 'd', 'a', 'b'];
application.output(a.lastIndexOf('b'));
application.output(a.lastIndexOf('b', 4));
application.output(a.lastIndexOf('z'));
```

map(callback)

Runs a function on every item in the array and returns the results in an array.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Object callback ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = ['a', 'b', 'c'];
var a2 = a.map(function(item) { return item.toUpperCase(); });
application.output(a2);
```

map(callback, thisObject)

Runs a function on every item in the array and returns the results in an array.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Object callback ;
Array thisObject ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var a = ['a', 'b', 'c'];
var a2 = a.map(function(item) { return item.toUpperCase(); });
application.output(a2);
```

pop()

Pops the last string off the array and returns it.

Returns

Object

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var lastword = words.pop();
```

push(value1)

Mutates an array by appending the given elements and returning the new length of the array.

Parameters

Object value1 ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete");
words.push("In","Out");
```

push(value1, value2)

Mutates an array by appending the given elements and returning the new length of the array.

Parameters

Object value1 ;
Object value2 ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete");
words.push("In","Out");
```

push(value1, value2, valueN)

Mutates an array by appending the given elements and returning the new length of the array.

Parameters

Object value1 ;
Object value2 ;
Object valueN ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete");
words.push("In","Out");
```

reduce(f, initialValue)

Reduces the array to a single value by executing a provided function for each value of the array (from left-to-right).

Parameters

Function	f	Function to execute on each element in the array, taking four arguments: -accumulator: accumulates the callback's return values; it is the accumulated value previously returned in the last invocation of the callback, or initialValue, if supplied (see below). -currentValue: the current element being processed in the array. -currentIndex (Optional): the index of the current element being processed in the array (starts at index 0, if an initialValue is provided, and at index 1 otherwise) -array (Optional): the array reduce() was called upon.
Object	initialValue	Value to use as the first argument to the first call of the callback. If no initial value is supplied, the first element in the array will be used.

Returns

Object Object

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var euros = [29.76, 41.85, 46.5];
var sum = euros.reduce( function(total, amount) {
  return total + amount
});
```

reverse()

Puts array elements in reverse order.

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
words.reverse();
```

shift()

Decreases array element size by one by shifting the first element off the array and returning it.

Returns

Object

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
words.shift();
```

slice(begin)

The slice method creates a new array from a selected section of an array.

Parameters

Object begin ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var nwords1 = words.slice(3, 5);
```

slice(begin, end)

The slice method creates a new array from a selected section of an array.

Parameters

Object begin ;
Object end ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var nwords1 = words.slice(3, 5);
```

some(callback)

Runs a function on items in the array while that function returns false. It returns true if the function returns true for any item it could visit.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function isNumber(value) { return typeof value == 'number'; }
var a1 = [1, 2, 3];
application.output(a1.some(isNumber));
var a2 = [1, '2', 3];
application.output(a2.some(isNumber));
```

some(callback, thisObject)

Runs a function on items in the array while that function returns false. It returns true if the function returns true for any item it could visit.
The callback function is invoked with three arguments: the element value, the element index, the array being traversed.

Parameters

Function callback ;
Array thisObject ;

Returns

Boolean

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
function isNumber(value) { return typeof value == 'number'; }
var a1 = [1, 2, 3];
application.output(a1.some(isNumber));
var a2 = [1, '2', 3];
application.output(a2.some(isNumber));
```

sort()

Sorts the array elements in dictionary order or using a compare function passed to the method.

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
words.sort();
```

sort(function)

Sorts the array elements in dictionary order or using a compare function passed to the method.

Parameters

Function function ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
words.sort();
```

splice(arrayIndex, length)

It is used to take elements out of an array and replace them with those specified.

Parameters

Object arrayIndex ;
Object length ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var nwords1 = words.splice(3, 2, "done", "On");
```

splice(arrayIndex, length, value1)

It is used to take elements out of an array and replace them with those specified.

Parameters

Object arrayIndex ;
Object length ;
Object value1 ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var nwords1 = words.splice(3, 2, "done", "On");
```

splice(arrayIndex, length, value1, value2)

It is used to take elements out of an array and replace them with those specified.

Parameters

Object arrayIndex ;
Object length ;
Object value1 ;
Object value2 ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var nwords1 = words.splice(3, 2, "done", "On");
```

splice(arrayIndex, length, value1, value2, valueN)

It is used to take elements out of an array and replace them with those specified.

Parameters

Object arrayIndex ;
Object length ;
Object value1 ;
Object value2 ;
Object valueN ;

Returns

Array

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("limit","lines","finish","complete","In","Out");
var nwords1 = words.splice(3, 2, "done", "On");
```

unshift(value1, value2, valueN)

Places element data at the start of an array.

Parameters

Object value1 ;
Object value2 ;
Object valueN ;

Returns

Number

Supported Clients

SmartClient,WebClient,NGClient,MobileClient

Sample

```
var words = new Array("finish","complete","In","Out");
words.unshift("limit","lines");
```