Microsoft AJAX Library: Array Type Extensions

Array.add (array, item) S

Adds an element to the end of an `Array` object.

```
var a = ['a','b','c','d'];
Array.add(a, 'e');
// a = ['a','b','c','d','e']
```

Array.addRange (array, items) S

Copies all the elements of a specified `Array` object to the end of the array.

```
var a = ['a', 'b', 'c', 'd', 'e'];
var b = ['f', 'g','h'];
Array.addRange(a, b);
// a = ['a','b','c','d','e','f','g','h']
```

Array.clear (array) S

Removes all elements from an Array instance.

Array.clone (array) S

Creates a shallow copy of an Array object.

Remarks

A shallow copy contains only the elements of the array, whether they are reference types or value types. However, it does not contain the referenced objects. The references in the new `Array` object point to the same objects as in the original `Array` object. In contrast, a deep copy of an `Array` object copies the elements and everything directly or indirectly referenced by the elements.

```
var a = ['a','b','c','d'];
var b = Array.clone(a);
// b = ['a','b','c','d']
```

Array.contains (array, item) S

Determines whether the specified object exists as an element in an `Array` object.

```
var a = ['red','green','blue','yellow'];
var b = Array.contains(a, "red");
// b = true
```

Array.dequeue (array) S

Removes the first element from the `Array` object and returns it.

```
var myArray = [],result = "";
Array.add(myArray, 'a');
Array.add(myArray, 'b');
Array.add(myArray, 'c');
Array.add(myArray, 'd');
result = Array.dequeue(myArray);
// myArray = ['b','c', 'd'], result = 'a'
```

Array.forEach (array, method, context) S

Performs a specified action on each element of an `Array` object. Skips elements in the array that have a value of `undefined`.

```
var a = ['a', 'b', 'c', 'd'];
a[5] = 'e';
var result = '';

function appendToString(arrayElement, index, array) {
    // "this" is the context parameter, i.e. '|'.
    result += arrayElement + this + index + ',';
}
Array.forEach(a, appendToString, '|');
// result = a|0,b|1,c|2,d|3,e|4,
```

Array.indexOf (array, item, start) S

Searches for the specified element of an `Array` object and returns its index. If `item` is not found in the array, returns -1.

```
var a = ['red', 'blue', 'green', 'blue'];
var myFirstIndex = Array.indexOf(a, "blue");
var mySecondIndex = Array.indexOf (a, "blue", (myFirstIndex + 1));
// myFirstIndex  = 1, mySecondIndex = 3
```

Array.insert (array, index, item) S

Inserts a single item into a specified index of an `Array` object.

```
var a = ['a', 'b', 'd', 'e'];
Array.insert(a, 2, 'c');
// a = ['a','b','c','d','e']
```

Array.parse (value) S

Creates an array from a string representation.

```
var a  = Array.parse ("['red', 'blue', 'green']");
// a[0] = 'red', a[1] = 'blue', a[2] = 'green'
```

Array.enqueue (array, item) S

Adds the specified object to the end of an `Array` object. See also `Array.dequeue ()`.

Array.remove(array, item) S

Removes the first occurrence of the specified item from an `Array` object.

```
var a = ['a', 'b', 'c', 'd', 'e'];
Array.remove(a, 'c');
// a = ['a','b','d','e']
Array.removeAt(a, 2);
// a = ['a','b','e']
```

Array.removeAt(array, index) S

Removes an element from an `Array` object at a specified index location.

S A function is static and is invoked without creating an instance of the object