Array

Array.prototype

Methods

static Array.clear() {Array}

AXUtil.js, line 1967

Array를 빈 Array 로 변경합니다.

Example

```
var a = [1,2,3];
trace(a);
// [1, 2, 3]
trace(a.clear());
// []
trace(a);
// []
```

static Array.convertTree(parentKey, childKey, hashDigit) {Object}

AXUtil.js, line 2299

리스트형 데이터를 부모 참조키와 자식 참조키를 이용하여 트리형 데이터로 변환처리 합니다.

Name	Type	Default	Description
parentKey	String		
childKey	String		
hashDigit	String	3	optional 트리의 주소값에 해당하는 hash 의 자릿수 단위 설정 (기본값 3)

Example

```
var a = [
    {pno:0, no:1, name:"장기영"},
    {pno:1, no:2, name:"장기영"},
    {pno:1, no:3, name:"장기영"},
    {pno:3, no:4, name:"장기영"},
    {pno:3, no:5, name:"장기영"},
    {pno:5, no:6, name:"장기영"},
    {pno:5, no:7, name:"장기영"}
];

var myTree = a.convertTree("pno", "no");
trace(myTree);
//[{"pno":0, "no":1, "name":"장기영", "subTree":[{"pno":1, "no":2, "name":"장기영", "__subTreeLength":0, "subTree":[], "pHash":"000_000", "hash":"000_000_000"}, {"pno":1, "no":3, "name":"장기영", "__subTreeLength":2, "subTree":[{"pno":3, "no":4,
```

```
"name":"장기영", "__subTreeLength":0, "subTree":[], "pHash":"000_000_001",
"hash":"000_000_001_000"}, {"pno":3, "no":5, "name":"장기영", "__subTreeLength":2, "
subTree":[{"pno":5, "no":6, "name":"장기영", "__subTreeLength":0, "subTree":[], "pHa
sh":"000_000_001_001", "hash":"000_000_001_001_000"}, {"pno":5, "no":7, "name":"장기
영", "__subTreeLength":0, "subTree":[], "pHash":"000_000_001_001",
"hash":"000_000_001_001_001"}], "pHash":"000_000_001", "hash":"000_000_001_001"}],
"pHash":"000_000", "hash":"000_000_001"}], "__subTreeLength":2, "pHash":"000", "has
h":"000_000"}]
```


static Array.first() {Object}

AXUtil.js, line 1986

Array의 첫번째 아이템을 반환합니다.

Example

```
1.  var a = [1,2,3];
2.  trace(a.first());
3.  // 1
4.
5.  var b = [{a:"액시스제이"}, 2, 3];
6.  trace(b.first());
7.  // {"a":"액시스제이"}
8.
9.  var c = [[1,2,3], 2, 3];
10. trace(c.first());
11. // [1, 2, 3]
```

static Array.getIndex(context) {Object}

AXUtil.js, line 2372

조건에 맞는 아이템을 index 값과 함께 반환합니다.

Name	Type	Description
context	function	

Example

```
var b = [1,2,3,4];
trace(b);
// [1, 2, 3, 4]
trace(b.getIndex(function(idx, item){
   return (this.item >= 3);
}));
//  {"item":3, "index":2}
```

static Array.getMaxObject(key) {Object}

AXUtil.js, line 2213

Object Array의 키를 정렬한후 가장 큰 값을 반환합니다.

Name	Type	Description
key	String	

Example

```
var myArray = [{a:2},{a:99},{a:1}];
myArray.getMaxObject("a");
// Object {a: 99}
```

---

static Array.getMinObject(key) {Object}

AXUtil.js, line 2190

Object Array의 키를 정렬한후 가장 작은 값을 반환합니다.

Name	Type	Description
key	String	

Example

```
var myArray = [{a:99},{a:2},{a:1}];
myArray.getMinObject("a");
// Object {a: 1}
```

---

static Array.getToSeq(seq) {Object}

AXUtil.js, line 2030

인자값에 해당하는 인덱스의 아이템을 반환합니다.

Name	Type	Description
seq	Number	

Example

```
var a = [1,2,3];
trace(a.getToSeq(1));
// 2

var a = [1,{a:2},3];
trace(a.getToSeq(1));
// {"a":2}
```

---

static Array.hasObject(callBack) {Object}

AXUtil.js, line 2153

사용자가 정의한 조건에 맞는 아이템을 한 개만 반환합니다.

Name	Type	Description
callBack	function	

Example

```
var a = [1,2,3,4];
trace(a);
// [1, 2, 3, 4]
trace(a.has(function(idx, item){
   return (item == 3);
}));
// 3

var b = [1,2,3,4];
trace(b);
// [1, 2, 3, 4]
trace(b.has(function(idx, item){
   return (this.item == 3);
}));
// 3
```

---

static Array.last() {Object}

AXUtil.js, line 2008

Array의 마지막 아이템을 반환합니다.

Example

```
var a = [1,2,3];
trace(a.last());
// 1

var b = [1, 2, {a:"액시스제이"}];
trace(b.last());
// {"a":"액시스제이"}

var c = [1, 2, [1,2,3]];
trace(c.last());
// [1, 2, 3]
```

---

static Array.remove(callBack) {Array}

AXUtil.js, line 2063

사용자가 정의한 조건에 맞는 아이템을 제거한 Array 를 반환합니다.

Name	Type	Description
callBack	function	remove 처리할 대상에 return true; 하면 true 인 대상이 제거 됩니다.

Example

```
var a = [1,2,3,4];
trace(a);
// [1, 2, 3, 4]
a = a.remove(function(idx, item){
   return (item == 3);
});
```

```
 trace(a);
 // [1, 2, 4]

 var b = [1,2,3,4];
 trace(b);
 // [1, 2, 3, 4]
 b = b.remove(function(){
    return (this.item == 3 || this.index == 0);
});
 trace(b);
 // [2, 4]
```

---

static Array.search(callBack) {Number}

AXUtil.js, line 2097

사용자가 정의한 조건에 맞는 아이템 갯수를 반환합니다.

Name	Type	Description
callBack	function	

Example

```
 var a = [1,2,3,4];
 trace(a);
 // [1, 2, 3, 4]
 trace(a.search(function(idx, item){
    return (item < 3);
}));
 // 2
```

---

static Array.searchObject(callBack) {Array}

AXUtil.js, line 2121

사용자가 정의한 조건에 맞는 아이템을 모두 반환합니다.

Name	Type	Description
callBack	function	

Example

```
 var a = [1,2,3,4];
 trace(a);
 // [1, 2, 3, 4]
 trace(a.searchObject(function(idx, item){
    return (item < 3);
}));
 // [1, 2]

 var b = [1,2,3,4];
 trace(b);
 // [1, 2, 3, 4]
 trace(b.searchObject(function(idx, item){
    return (this.item < 3);
```

```
}));
 // [1, 2]
```

Documentation generated by JSDoc 3.2.2 on Sun Nov 30 2014 00:50:15 GMT+0900 (KST)