ARRAYS

TBC

📖 REVIEW

📜 Arrays: core

👷 Methods for creating arrays

🧙 Mutable methods

👨‍💻 Immutable methods

🧐 Methods for searching

🚴 Methods for iterating

TBC

📖 ARRAY

Data structure for storing list of values, identifier by index.

TBC

📖 CREATING OF ARRAY

```
1 var arr = [1,2,3];
2
```

TBC

📖 CREATING OF ARRAY

```
1 var arr = new Array(1,2,3);
2
```

TBC

📖 ARRAY TYPE

```
1 typeof [1,2,3];
2
```

📖 RIGHT CHECK

```
1 Array.isArray([1,2,3]);
2
```

TBC

📖 ARRAY LENGTH

```
1 [1,2,3].length;
2
```

📖 ARRAY READING

```
1 var arr = [1,2,3];
2 var first = arr[0];
3
```

TBC

📖 ARRAY WRITING

```
1 var arr = [1,2,3];
2 arr[0] = 7;
3
```

TBC

📖 METHODS FOR CREATING

- 🔧 of
- 🔧 from
- 🔧 split

📖 OF

```
1 var arr = Array.of(7);
2
```

📖 FROM

```
1 var arr = Array.from('Content');
2
```

📖 SPLIT

```
1 var names = 'Alex, Pitter, Oliver, Mark';
2
3 var arr = names.split(', ');
```

TBC

📖 MUTABLE METHODS

- 🔧 fill
- 🔧 pop
- 🔧 push
- 🔧 reverse
- 🔧 sort
- 🔧 splice
- 🔧 shift
- 🔧 unshift
- 🔧 copyWithin

TBC

📖 FILL

```
1 var arr = [1,2,3];
2 arr.fill(6);
3
```

TBC

📖 POP

```
1 var arr = [1,2,3];
2 arr.pop();
3
```

TBC

📖 PUSH

```
1 var arr = [1,2,3];
2 arr.push(4,5);
3
```

TBC

📖 REVERSE

```
1 var arr = [1,2,3];
2 arr.reverse();
3
```

TBC

📖 SORT

```
1 var arr = [3,2,7,1];
2 arr.sort();
3
```

TBC

📖 SPLICE

```
1 var arr = [1,3,4];
2 arr.splice(1,0,2);
3
```

TBC

📖 SHIFT

```
1 var arr = [1,2,3];
2 arr.shift();
3
```

📖 UNSHIFT

```
1 var arr = [2,3];
2 arr.unshift(1);
3
```

📖 COPYWITHIN

```
1 var arr = [1,2,3,0];
2 arr.copyWithin(1,3); // [1, 0, 3, 0]
3
```

TBC

📖 IMMUTABLE METHODS

- 🔧 concat
- 🔧 flat
- 🔧 flatMap
- 🔧 join
- 🔧 slice

TBC

📖 CONCAT

```
1 var first = [1,2,3];
2 var second = [4,5,6];
3 var arr = first.concat(second);
4
```

TBC

📖 FLAT

```
1 var source = [1, 2, [3, 4]];
2 var destination = source.flat();
3
```

TBC

📖 FLATMAP

```
1 var source = [1,2,3];
2 var destination = source.flatMap(function(item){
3   return [[item * 2]];
4 });
5
```

TBC

📖 JOIN

```
1 var source = [1,2,3];
2 var str = source.join(' ');
3
```

📖 SLICE

```
1 var source = [1,2,3,4];
2 var destination = source.slice(1,4);
3
```

TBC

📖 METHODS FOR SEARCHING

- 🔧 find
- 🔧 findIndex
- 🔧 indexOf
- 🔧 lastIndexOf

TBC

📖 FIND

```
1 var arr = [1,2,3,4,5];
2
3 var item = arr.find(function(element) {
4   return element > 2;
5 });
6
```

TBC

📖 FINDINDEX

```
1 var arr = [1,2,3,4,5];
2
3 var item = arr.findIndex(function(element) {
4   return element > 2;
5 });
6
```

📖 INDEXOF

```
1 var arr = [1,2,3,4,5];
2
3 var item = arr.indexOf(3);
4
```

TBC

📖 LASTINDEXOF

```
1 var arr = [1,2,3,4,3];
2
3 var item = arr.lastIndexOf(3);
4
```

TBC

📖 METHODS FOR ITERATING

- 🔧 forEach
- 🔧 filter
- 🔧 map
- 🔧 reduce
- 🔧 reduceRight
- 🔧 every
- 🔧 some

TBC

📖 FOREACH

```
1 var arr = ['Alex', 'Pitter', 'Oliver'];
2
3 arr.forEach(function(item, i, arr) {
4   console.log( i + ': ' + item + ' (array:' + arr + ')' );
5 });
```

TBC

📖 FILTER

```
1 var arr = [1, -1, 2, -2, 3];
2
3 var positiveArr = arr.filter(function(number) {
4   return number > 0;
5 });
```

TBC

📖 MAP

```
1 var arr = ['HTML', 'CSS', 'JavaScript'];
2
3 var transformed = arr.map(function(name) {
4   return name.toLowerCase();
5 });
```

TBC

📖 REDUCE

```
1 var arr = [1, 2, 3];
2
3 var result = arr.reduce(function(sum, current) {
4   return sum + current;
5 }, 0);
```

TBC

📖 REDUCERIGHT

```
1 var arr = ['a', 'b', 'c'];
2
3 var result = arr.reduceRight(function(sum, current) {
4   return sum + current;
5 }, '');
```

TBC

📖 EVERY

```
1 var arr = [1, -1, 2, -2, 3];
2
3 var status = arr.every(function(n) {
4   return n > 0;
5 });
```

TBC

📖 SOME

```
1 var arr = [1, -1, 2, -2, 3];
2
3 var status = arr.some(function(n) {
4   return n > 0;
5 });
```

TBC

HERE GOES!

TBC