

# JavaScript

## ΜΑΘΗΜΑ 6.2

### ΜΕΘΟΔΟΙ ΠΙΝΑΚΩΝ

#### ΠΕΡΙΕΧΟΜΕΝΑ:

1. Μέθοδοι Πινάκων
  1. Αναζήτηση και Ταξινόμηση
  2. Εισαγωγή/Απομάκρυνση στα Άκρα
  3. Υποπίνακες με τη slice()
  4. Η μέθοδος splice()
  5. Άλλες Μέθοδοι Πινάκων
2. Τυχαίοι Αριθμοί

Γεώργιος Κ.

Σμαραγδένιος Χορηγός Μαθήματος

Εμμανουήλ Α.

Σμαραγδένιος Χορηγός Μαθήματος

### Αναζήτηση:

- Μέθοδοι που αναζητούν ένα στοιχείο στον πίνακα:

Μέθοδος	Επεξήγηση
<code>indexOf(elem[, n])</code>	Επιστρέφει τη θέση της πρώτης εμφάνισης του <code>elem</code> στον πίνακα
<code>lastIndexOf(elem[, n])</code>	Επιστρέφει τη θέση της τελευταίας εμφάνισης του <code>elem</code> στον πίνακα

- Επιστρέφει -1 αν δεν υπάρχει το στοιχείο
- Το προαιρετικό 2<sup>ο</sup> όρισμα καθορίζει τη θέση από την οποία θα ξεκινήσει η αναζήτηση
- Ο έλεγχος γίνεται με την αυστηρή ισότητα
- Επίσης ορίζεται η:

Μέθοδος	Επεξήγηση
<code>includes(elem)</code>	true/false, αν το <code>elem</code> περιέχεται στον πίνακα

### Παράδειγμα 1: search

```
let array = [1, 2, 3, 4, 5];
console.log(array.indexOf(3))
console.log(array.lastIndexOf(4))
console.log(array.includes(3))
console.log(array.indexOf(3, 4));
console.log(array.lastIndexOf(2, -4));
```

### Ταξινόμηση:

- Χρησιμοποιούμε τη μέθοδο:

Μέθοδος	Επεξήγηση
<code>sort()</code>	Ταξινομεί σε αύξουσα σειρά τον πίνακα

- συνδυαστικά με τη `sort()` μπορούμε να χρησιμοποιήσουμε την:

Μέθοδος	Επεξήγηση
<code>reverse()</code>	Αντιστρέφει τον πίνακα

### Παράδειγμα 2: sort

```
let array = [5, 1, 2, 4, 3];
array.sort();
console.log(array);
let array2 = Array.from(array);
array2.reverse();
console.log(array2);
```

### Σημείωση:

- Υπάρχει και παραλλαγή της `sort()` που δέχεται ως όρισμα έναν συγκριτή, με τον οποίο θα ταξινομήσει τα στοιχεία του πίνακα
- Ο συγκριτής (comparator) είναι μια συνάρτηση που με όρισμα δύο στοιχεία `a, b` επιστρέφει:
  - <0, αν `a < b`
  - =0, αν `a == b`
  - >0, αν `a > b` (θα το μελετήσουμε στο μάθημα των συναρτήσεων)

### Εισαγωγή/Διαγραφή στα άκρα:

- Εισάγουμε/Διαγράφουμε στοιχεία στο τέλος του πίνακα:

Μέθοδος	Επεξήγηση
push(elem)	Προσθέτει το elem στο τέλος του πίνακα
push(e1,..., en)	Προσθέτει τα στοιχεία στο τέλος του πίνακα
pop()	Αφαιρεί και επιστρέφει το τελευταίο στοιχείο του πίνακα (επιστρ. undefined σε κενό πίνακα)

- Εισάγουμε/Διαγράφουμε στοιχεία στην αρχή του πίνακα:

Μέθοδος	Επεξήγηση
unshift(elem)	Προσθέτει το elem στην αρχή του πίνακα
unshift(e1,..., en)	Προσθέτει τα στοιχεία στην αρχή του πίνακα
shift()	Αφαιρεί και επιστρέφει το πρώτο στοιχείο του πίνακα (επιστρ. undefined σε κενό πίνακα)

### Παράδειγμα 3: stack

```
// stack
let stack = [];
stack.push(1);
stack.push(2);
let elem = stack.pop();
console.log(elem);
stack.push(3);
while (stack.length>0)
    console.log(stack.pop());
```

- Συναφής είναι και η μέθοδος:

Μέθοδος	Επεξήγηση
concat(array2)	Επιστρέφει νέο πίνακα που περιέχει τη συνένωση του πίνακα με τον array2

### Παράδειγμα 4: concat example

```
let array = [1, 2, 3];
let array2 = [4, 5];
let array3 = array.concat(array2);
console.log(array + " | " + array2 + " | " + array3);
array.push(...array2);
console.log(array + " | " + array2);
```

### Άσκηση 1:

Υλοποιήστε μια ουρά με πίνακα (η εισαγωγή γίνεται στο ένα άκρο και η απομάκρυνση στοιχείου από το άλλο) και κάνετε τις πράξεις:

- Εισαγωγή του 1,
- Εισαγωγή του 2,
- Απομάκρυνση στοιχείου (και εκτύπωσή του)
- Εισαγωγή του 3
- Εισαγωγή του 4
- Απομάκρυνση στοιχείου (και εκτύπωσή του)
- Εισαγωγή του 5
- Απομάκρυνση και εκτύπωση των υπόλοιπων στοιχείων

Υποπίνακας με τη slice():

- Την χρησιμοποιούμε για να κατασκευάσουμε έναν υποπίνακα του αρχικού (χωρίς να επηρεάζουμε τον αρχικό πίνακα):

Μέθοδος	Επεξήγηση
slice()	Αντίγραφο του πίνακα
slice(start)	Υποπίνακας των θέσεων start,...,length-1
slice(start, end)	Υποπίνακας των θέσεων start,..., end-1

- Σημείωση: Τα ορίσματα μπορούν να έχουν και αρνητικές τιμές (οπότε υπολογίζονται από το τέλος, π.χ. -1 γίνεται length-1, -2 γίνεται length-2 κ.ο.κ.)

Παράδειγμα 5: slice

```
// init
let array = [];
for (let i=0; i<100; i++)
    array.push(i);

// make subarrays
for (let i=0; i<20; i++) {
    console.log(array.slice(5*i, 5*i+5));
}
```

Αρνητικοί Δείκτες:

- Επιτρέπονται ως ορίσματα αρνητικοί δείκτες (η αρίθμηση της τελευταίας θέσης είναι -1)

Παράδειγμα 6: slice negative indexes

```
let array = [0,1,2,3,4,5,6];

console.log(array.slice(-2));
console.log(array.slice(1, -1));
console.log(array.slice(-5, -2));
console.log(array.slice(5, -5));
```

Άσκηση 2: (βλ. υποχρεωτικά και βίντεο)

- Μελετήστε το πρόγραμμα: exercise02\_initial.html
- Παρατηρήστε ότι μετά από την εφαρμογή της slice(), όταν επηρεάζουμε τον νέο πίνακα, επηρεάζεται και ο αρχικός πίνακας.
- Πως μπορούμε να διορθώσουμε αυτό το πρόβλημα;

### Η μέθοδος splice():

- Αφαιρεί στοιχεία από τον πίνακα και τα επιστρέφει σε έναν νέο πίνακα (Προσοχή: ο αρχικός πίνακας επηρεάζεται):

Μέθοδος	Επεξήγηση
splice(start)	Αφαιρεί και επιστρέφει τα στοιχεία των θέσεων start, .., length-1
splice(start, cnt)	Αντίστοιχα των θέσεων start, ..., start+cnt-1

- Επίσης επιτρέπονται αρνητικές τιμές στους δείκτες με την ίδια λειτουργία, όπως στη slice().
- Μπορούμε να προσθέσουμε στοιχεία στη θέση αυτών που αφαιρούνται (στον αρχικό πίνακα) με την παραλλαγή:

Μέθοδος	Επεξήγηση
splice(start, cnt, item1, ..., itemN)	Αφαιρεί τα στοιχεία στις θέσεις [start, start+cnt) Προσθέτει τα στοιχεία item1,...,itemN ξεκινώντας από τη θέση start. Επιστρέφει τον πίνακα με τα στοιχεία που αφαιρέθηκαν

### Παράδειγμα 5: splice remove

```
let array = [];
for (let i=0; i<100; i++)
    array.push(i);

console.log(array.splice(50));
console.log(array.splice(10,10));
console.log(array);
```

### Παράδειγμα 6: splice replace

```
let array = [1,2,3,4,5];

console.log(array.splice(3));
console.log(array);

array = [1,2,3,4,5]
console.log(array.splice(1,3,11,12,13,14));
console.log(array);
```

### Παράδειγμα 7: splice insert

```
let array = [1,2,3,4,5];

console.log(array.splice(1,0,11,12,13));
console.log(array);
```

### Άσκηση 3 (διαφορά splice και delete):

- Κατασκευάστε έναν πίνακα με 5 στοιχεία.
- Διαγράψτε το στοιχείο στη θέση 0 με τη delete
- Διαγράψτε το στοιχείο στη θέση 2 με τη splice
- Εκτυπώστε τον τελικό πίνακα

### Γέμισμα Πίνακα:

- Για να γεμίσουμε τον πίνακα με τιμές:

Μέθοδος	Επεξήγηση
fill(value)	Όλες οι θέσεις του πίνακα αρχικοποιούνται με την τιμή value
fill(value, start)	Γεμίζει τις θέσεις [start, length) με τη value
fill(value, start, end)	Γεμίζει τις θέσεις [start, end) με τη value

### Αντιγραφή μέρους του πίνακα στον εαυτό του:

- Η copyWithin(), σε δύο παραλλαγές:

Μέθοδος	Επεξήγηση
copyWithin(index, start)	Αντιγράφει το a[start]->a[index],..., a[length]->a[index+length-1]
copyWithin(index, start, end)	Αντιγράφει το a[start]->a[index],..., a[end]->a[index+end-1]

### Μετατροπή σε Συμβολοσειρά:

Μέθοδος	Επεξήγηση
join(delimiter)	Ενώνει τα στοιχεία του πίνακα με τη συμβολοσειρά delimiter
toString()	Μετατρέπει τον πίνακα σε μία συγκεκριμένη μορφή συμβολοσειράς

- Ως delimiter στην join μπορούμε να βάλουμε οποιαδήποτε συμβολοσειρά διαχωρισμού, π.χ. [1,2,3].join("--")=="1--2--3"

### Παράδειγμα 7: fill join copywithin

```
// fill
let array = [1, 2, 3, 4, 5];
array.fill(0);
array.fill(1, 1);
array.fill(2, 2, 4);
console.log(array);

// copyWithin
array = [0, 1, 2, 3, 4];
array.copyWithin(1, 3);
console.log(array);

array = [0, 1, 2, 3, 4];
array.copyWithin(1, 2, 4);
console.log(array);

// join
array = [1, 2, 3, 4, 5];
console.log(array.join(", "))
console.log(array.join("--+--"))

// toString()
console.log(array.toString());
```

- Η JavaScript περιλαμβάνει «έτοιμο» κώδικα που κάνει τη ζωή μας πιο εύκολη.
- Έτσι η στατική μέθοδος `Math.random()` επιστρέφει έναν πραγματικό αριθμό στο `[0,1)`
- Δεν υπάρχει κάποια μέθοδος που να επιστρέφει τυχαίο ακέραιο, αλλά μπορούμε να γράψουμε μια δική μας συνάρτηση (επόμενο μάθημα) που κάνει αυτή τη λειτουργία (βλ. παράδειγμα `random`):

```
function randomFloat(min, max) { // [min,max)
  return Math.random() * (max - min) + min;
}
function randomInt(min, max) { // [min,max)
  return Math.floor(Math.random() * (max - min) + min);
}
```

**Άσκηση 4:**

[Συμβουλευθείτε το βίντεο]

Δηλώστε έναν πίνακα 10 θέσεων και αρχικοποιήστε τον με τυχαίους ακραίους στο εύρος 0-100. Έπειτα:

- Τυπώστε τα περιεχόμενα του πίνακα
- Υπολογίστε και τυπώστε τον μέσο όρο των στοιχείων του πίνακα.

**Άσκηση 5:**

Δηλώστε έναν πίνακα 10x40x9 θέσεων και αρχικοποιήστε τον με τυχαίους ακραίους στο εύρος `[0,5000]`.

- Τυπώστε τα περιεχόμενα του πίνακα.
- Εξετάστε αν υπάρχει το στοιχείο 1123 στον πίνακα.