

# Arrays

## Processing Sequences of Elements

---

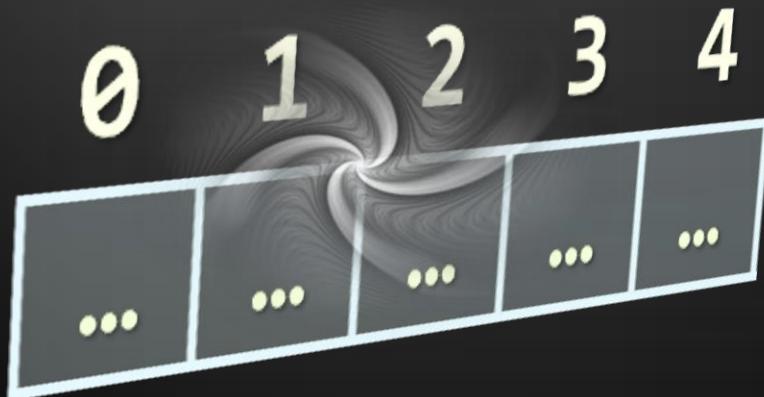


# Table of Contents

1. Declaring and Creating Arrays
2. Accessing Array Elements
3. Processing Array Elements
4. Dynamic Arrays
5. Sorting Arrays

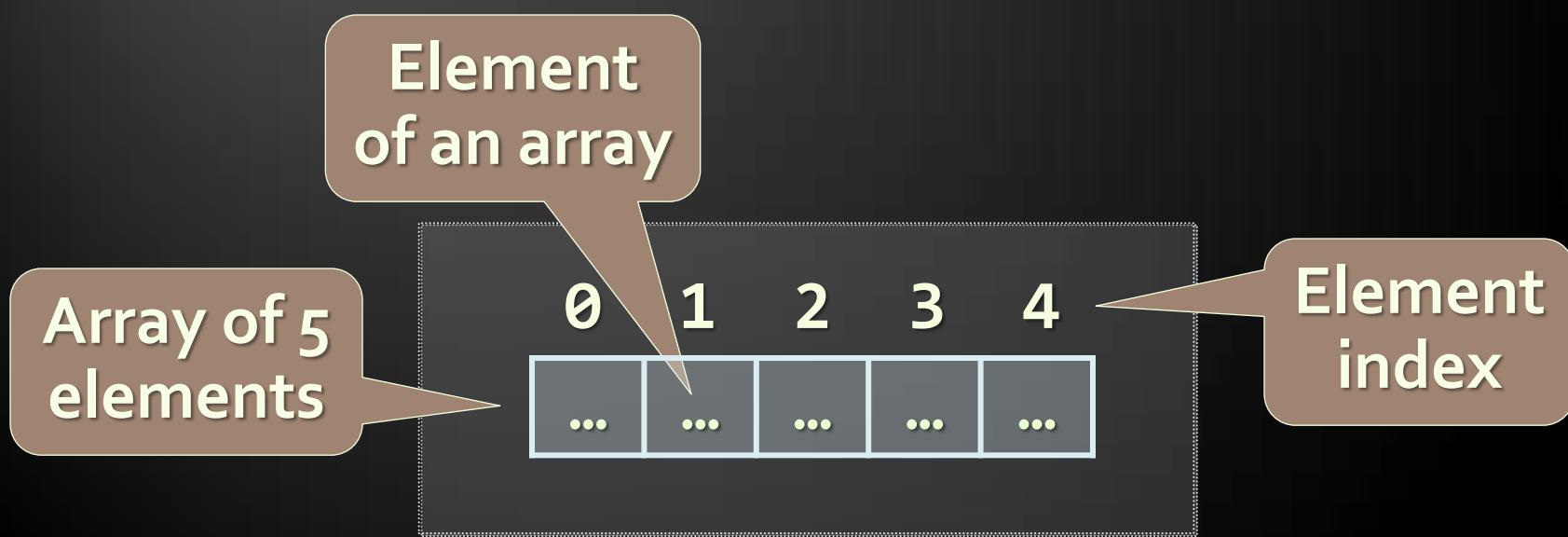


# Declaring and Creating Arrays



# What are Arrays?

- ◆ An array is a sequence of elements
  - ◆ The order of the elements is fixed
  - ◆ Does not have fixed size
  - ◆ Can get the current length (`Array.length`)



- ◆ Declaring an array in JavaScript (JS is typeless)

```
// Array holding integers
var numbers = [1, 2, 3, 4, 5];

// Array holding strings
var weekDays = ['Monday', 'Tuesday', 'Wednesday',
    'Thursday', 'Friday', 'Saturday', 'Sunday']

// Array of different types
var mixedArr = [1, new Date(), 'hello'];

// Array of arrays (matrix)
var matrix = [
    ['0,0', '0,1', '0,2'],
    ['1,0', '1,1', '1,2'],
    ['2,0', '2,1', '2,2']]
```

# Declare and Initialize Arrays

- ◆ Initializing an array in JavaScript can be done in three ways:
  - ◆ Using new Array(elements):

```
var arr = new Array(1, 2, 3, 4, 5);
```

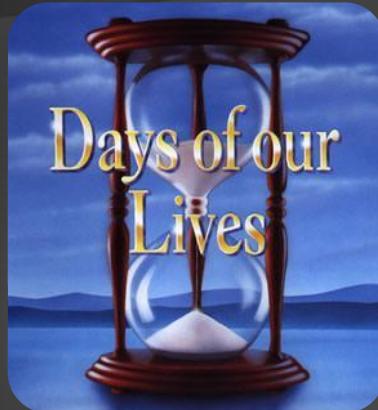
- ◆ Using new Array(initialLength):

```
var arr = new Array(10);
```

- ◆ Using array literal (recommended):

```
var arr = [1, 2, 3, 4, 5];
```





# Creating Arrays

Live Demo



# Accessing Array Elements

Read and Modify Elements by Index



# How to Access Array Element?

- ◆ Array elements are accessed using the square brackets operator [ ] (indexer)
  - Array indexer takes element's index as parameter in the range 0 ... length-1
  - The first element has index 0
  - The last element has index length-1
- ◆ Array elements can be retrieved and changed by the [ ] operator

# Reversing an Array – Example

- ◆ Reversing the elements of an array

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

// Get array size
var length = array.length; // 5

// Declare and create the reversed array
var reversed = new Array(length);

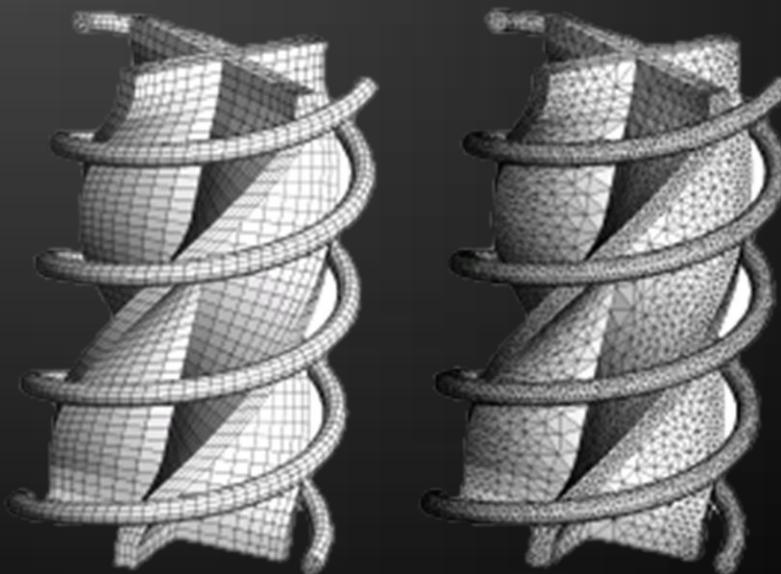
// Initialize the reversed array
for (var index = 0; index < length; index++) {
    reversed[length - index - 1] = array[index];
}
```

# Reversing an Array

Live Demo



# Processing Array Elements Using for and for-in



# Processing Arrays: for Statement

- ◆ Use for loop to process an array when
  - Need to keep track of the index
- ◆ In the loop body use the element at the loop index (array[index]):

```
for (var index = 0; index < array.length; index++) {  
    squares[index] = array[index] * array[index];  
}
```

# Processing Arrays Using for Loop – Examples

- ◆ Printing array of integers in reversed order:

```
var array = [1, 2, 3, 4, 5];
for (var i = array.length-1; i >= 0; i--) {
    console.log(array[i]);
}
// Result: 5 4 3 2 1
```

- ◆ Initialize all array elements with their corresponding index number:

```
for (var index = 0; index < array.length; index++) {
    array[index] = index;
}
```

# Processing Arrays: for-in

- ◆ How for-in loop works?

```
for (var i in array)
```

- ◆ i iterates through all the indexes of array
- ◆ Used when the indexes are unknown
  - ◆ All elements are accessed one by one
  - ◆ Order is not guaranteed



# Example: Processing Arrays Using for-in Loop

- ◆ Print all elements of an array of strings:

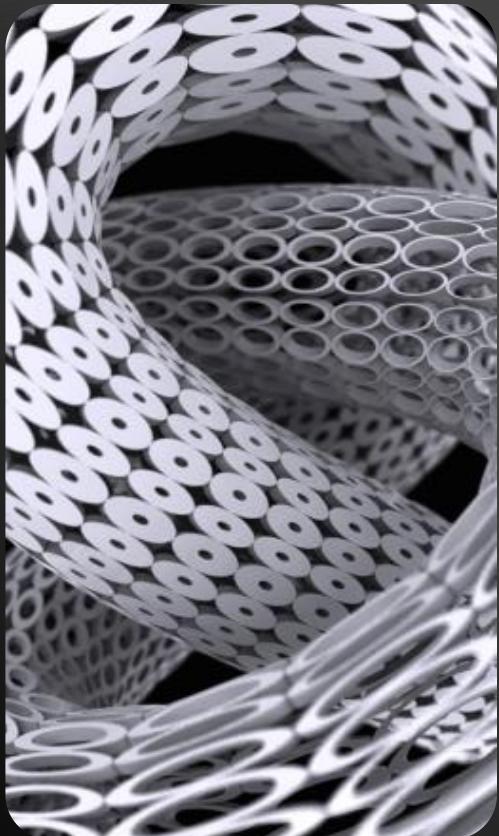
```
var capitals = [
    'Sofia',
    'Washington',
    'London',
    'Paris'
];

for (var i in capitals) {
    console.log(capitals[i]);
}
```



# Processing Arrays

Live Demo



# Dynamic Arrays



- ◆ All arrays in JavaScript are dynamic
  - ◆ Their size can be changed at runtime
  - ◆ New elements can be inserted to the array
  - ◆ Elements can be removed from the array
- ◆ Methods for array manipulation:
  - ◆ **array.push(element)**
    - ◆ Inserts a new element at the tail of the array
  - ◆ **array.pop()**
    - ◆ Removes the element at the tail
    - ◆ Returns the removed element

# Dynamic Arrays (2)

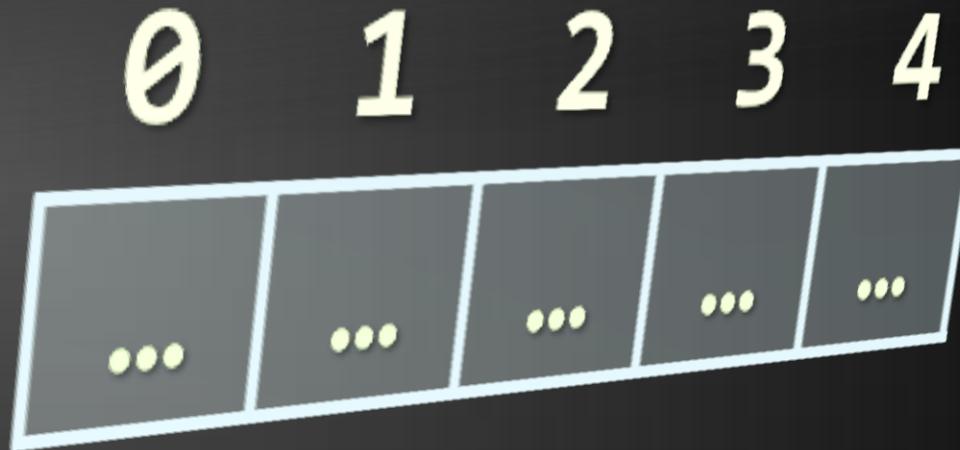
- ◆ Methods for array manipulation (cont.)
  - ◆ **array.unshift(element)**
    - ◆ Inserts a new element at the head of the array
  - ◆ **array.shift()**
    - ◆ Removes and returns the element at the head

```
var numbers = [1, 2, 3, 4, 5];
console.log(numbers.join('|')); // result: 1|2|3|4|5

var tail = number.pop();      // tail = 5;
console.log(numbers.join('|')); // result: 1|2|3|4

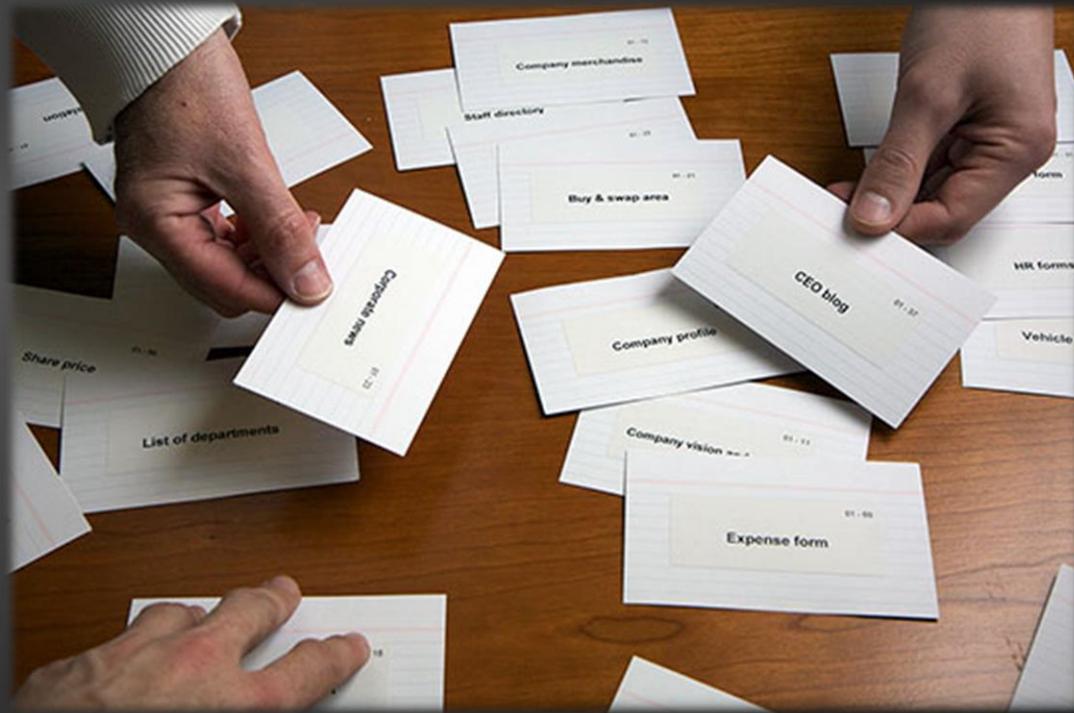
number.unshift(0);
console.log(numbers.join('|')); // result: 0|1|2|3|4

var head = number.shift();    // head = 0;
console.log(numbers.join('|')); // result: 1|2|3|4
```



# Dynamic Arrays

Live Demo



# Sorting Arrays

`Array.sort()` and `Array.sort(orderBy)`

- ◆ **array.sort()**

- ◆ Sorts the elements of the array ascending

```
var numbers = [5, 4, 2, 3, 1, 4, 5, 6, 7];
numbers.sort();
console.log(numbers.join(', '));
// result: 1, 2, 3, 4, 4, 5, 5, 6, 7
```

- ◆ Keep in mind that **array.sort** uses the string representation of the elements!
  - ◆ i.e. the number 5 is compared as the string "5"

```
var numbers = [5, 4, 23, 2];
numbers.sort();
console.log(numbers.join(', ')); // result: 2, 23, 4, 5
```

- ◆ Not quite sorted, right?

# Sorting Arrays with Compare Function

## ◆ `array.sort(compareFunc)`

- Sorts element using a compare function
- The compare function defines the sorting rules
  - Return negative or 0 to leave the elements
  - Return positive to swap them

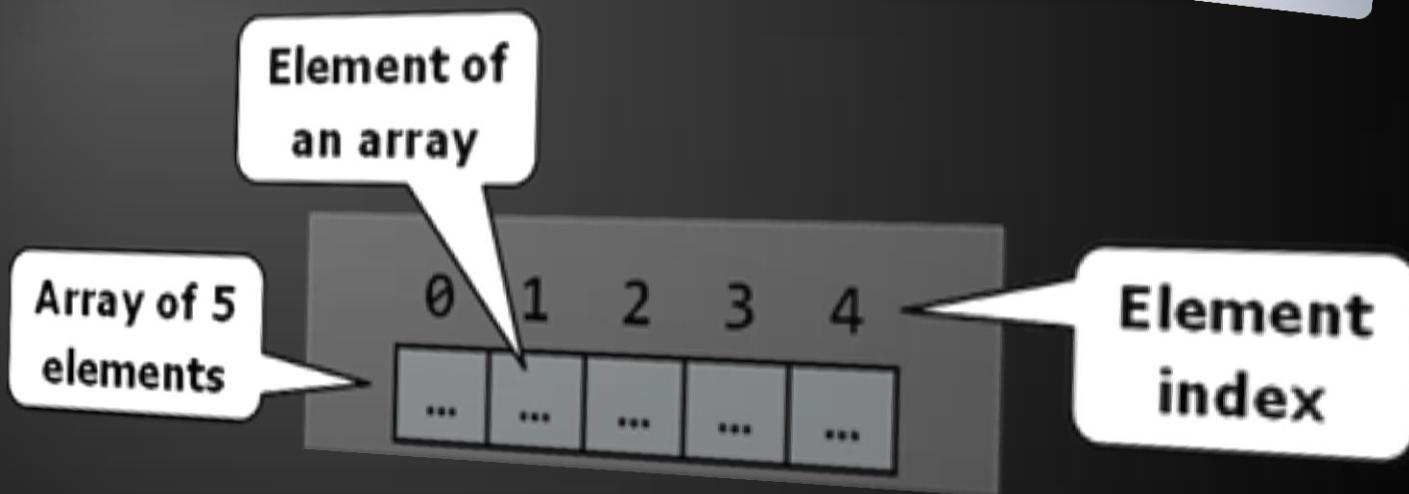
```
function orderBy(a, b) {  
    return (a == b) ? 0 : (a > b) ? 1 : -1;  
};  
var numbers = [5, 4, 23, 2];  
  
numbers.sort(orderBy);  
console.log(numbers.join(', '));  
// returns 2, 4, 5, 23
```



# Sorting Arrays

Live Demo

$$f(x, y) = x + y$$



## Other Array Functions

# Other Array Functions

- ◆ **array.reverse()**
  - ◆ Returns a new arrays with elements in reversed order
- ◆ **array.splice(index, count, elements)**
  - ◆ Adds and / or removes elements from an array
- ◆ **array.concat(elements)**
  - ◆ Inserts the elements at the end of the array and returns a new array
- ◆ **array.join(separator)**
  - ◆ Concatenates the elements of the array

# Other Array Functions (2)

- ◆ **array.filter(condition)**
  - ◆ Returns a new array with the elements that satisfy condition
- ◆ **array.forEach(function(item){})**
  - ◆ Iterates through the array and executes the function for each item

# Other Functions

Live Demo

# Other Array Functions (2)

- ◆ **array.indexOf(element)**
  - ◆ Returns the index of the first match in the array
  - ◆ Returns -1 if the element is not found
- ◆ **array.lastIndexOf(element)**
  - ◆ Returns the index of the last match in the array
  - ◆ Returns -1 if the element is not found
- ◆ **indexOf() and lastIndexOf() do not work in all browsers**
  - ◆ Need to add a predefined functionality

# indexOf() and lastIndexOf()

Live Demo

# Other Arrays Functions

- ◆ Arrays official documentation:
  - [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array)
- ◆ Checking for array
  - `typeof([1, 2, 3])` → object
  - `Array.isArray([1, 2, 3])` → true
  - Supported on all modern browsers



# Questions?



1. Write a script that allocates array of 20 integers and initializes each element by its index multiplied by 5. Print the obtained array on the console.
2. Write a script that compares two char arrays lexicographically (letter by letter).
3. Write a script that finds the maximal sequence of equal elements in an array.

Example: {2, 1, 1, 2, 3, 3, **2, 2, 2, 1**} → {2, 2, 2}.

4. Write a script that finds the maximal increasing sequence in an array. Example:  
 $\{3, \boxed{2}, 3, 4, 2, 2, 4\} \rightarrow \{2, 3, 4\}$ .
5. Sorting an array means to arrange its elements in increasing order. Write a script to sort an array. Use the "selection sort" algorithm: Find the smallest element, move it at the first position, find the smallest from the rest, move it at the second position, etc.  
Hint: Use a second array
6. Write a program that finds the most frequent number in an array. Example:  
 $\{\boxed{4}, 1, \boxed{1}, \boxed{4}, 2, 3, \boxed{4}, \boxed{4}, 1, 2, \boxed{4}, 9, 3\} \rightarrow 4 \text{ (5 times)}$

7. Write a program that finds the index of given element in a sorted array of integers by using the binary search algorithm (find it in Wikipedia).