**נושאים נוספים לקרוא:**

* Strict mode  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode>
* Why eval() is the evil
* Working with arrays
  + What will be printed to the console?

var person = [];  
person["firstName"] = "John";

person.push({kaki:”pipi”});  
console.log(person[0]);  
console.log(person.length)`

* Default params in functions
* Continue and Break statements  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/continue>   
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/break>
* Arrow Function and its differences from function()  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions>
* Closures
* Hoisting  
  <https://www.w3schools.com/js/js_hoisting.asp>
* Bind / Apply / Call
* Spread operator <https://developer.mozilla.org/he/docs/Web/JavaScript/Reference/Operators/Spread_operator>
* Destructuring in objects, arrays and in assignments

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Destructuring_assignment>

* .prototype
* Working asynchronously
  + With native promises
  + With async & await
* Labels  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/label>
* Creating a user-defined object (new)  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/new>
* New.target   
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/new.target>
* Classes  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/class>
* delete and the prototype chain  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/delete>
* in operator  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/in>
* Yield  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/yield>
* function\* as the new generator  
  [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function\*](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/function*)
* Object.assign
* Object new() VS Object.create()  
  <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Object_initializer>
* Template Literals & multi-line strings
* You should know these libraries, they are very cool and helpful.
  + Lodash
  + Underscore
  + Ramda
  + Immutable

**שאלות מסכמות:**

1. איזה סוגי אובייקטים קיימים בשפה? מה השימוש לכל אחד מהם?
2. מהם Truthy & Falsy values? למה לא כדאי להשתמש בהם?
3. מה זה null? מה זה NaN? מה זה undefined? מה ההבדל ביניהם?
4. מה זה Hoisting?
5. מה זה this? איך הוא מוגדר?
6. Closures – מה זה? מתי ואיך משתמשים בזה?
7. מה זה bind? מה זה apply? מה זה call? מתי נשתמש בכל אחד מהם?
8. מה ההבדל בין = לבין == לבין ===?
9. מה זה ecma-script ואיך הוא שונה מjs?
10. מה זה var? מה זה let? מה זה const?
11. מה זה object assign? מה זה spread operator? איך הם שונים זה מזה?
12. איך כותבים תנאי if בשורה אחת? (ternary operator)
13. מה הן הפונקציות map / filter / reduce / foreach ?
14. מה הם default params בפונקציות?
15. מה הם arrow functions? במה הם שונים מfunction רגיל?
16. איך יוצרים class בJS?
17. מה זה prototype של אובייקט?
18. מה זה promise? מתי נשתמש בזה?
19. מה זה then? מה זה async/await? במה הם שונים?
20. מה יקרה בשורת הקוד הבאה: const fieldText = userInput ?? "please enter your input”
21. מה יקרה בשורת הקוד הבאה: let result = data?.key1?.key2;