

מדריך JavaScript מתקדם ומקיף לבינה מלאכותית
Advanced and Comprehensive JavaScript Training Guide for AI #

מבוא מורחב ל-JavaScript

מה זה JavaScript ולמה זה חשוב?
JavaScript הוא שפת תכנות דינמית, מפורשת (interpreted) ורב-פרדיגמטית שפותחה במקור לדפדפנים אך כיום רצה גם בשרתים (Node.js), אפליקציות מובייל ועוד.

היסטוריה של JavaScript:
- **1995**: ברנדן אייך יוצר את JavaScript ב-10 יום בנטסקייפ
- **1997**: ECMAScript 1 - הסטנדרט הראשון
- **1999**: ES3 - הוספת regex, try/catch, better string handling
- **2009**: ES5 - strict mode, JSON, array methods
- **2015**: ES6/ES2015 - מהפכה: classes, modules, arrow functions, let/const
- **2016-היום**: עדכונים שנתיים עם תכונות חדשות

עקרונות הליבה של JavaScript

1. Dynamic Typing - טיפוסים דינמיים
javascript``
// החשתנה יכול לשנות טיפוס בזמן ריצה
let variable = 42; // number
variable = "Hello"; // string
variable = true; // boolean
variable = null; // null
variable = undefined; // undefined
variable = {}; // object
(variable = []); // object (array
variable = function() {}; // function

// בדיקת טיפוס
"console.log(typeof variable); // "function"

2. First-Class Functions - פונקציות כאזרחים ראשונים
javascript``
// פונקציות הן ערכים שניתן להעביר, לשמור ולהחזיר
(const myFunction = function(name
;`{return `Hello \${name
;}

// פונקציה כפרמטר
(function executeFunction(fn, value
;(return fn(value
{

;("const result = executeFunction(myFunction, "World

// פונקציה שמחזירה פונקציה
(function createMultiplier(factor
{ (return function(number
;return number * factor
;{
{

```
;(const double = createMultiplier(2
  console.log(double(5))); // 10
  ``
```

3.3 Prototype-Based Inheritance - ירושה מבוססת אב-טיפוס

```
javascript``
// כל אובייקט יש לו prototype
} = const person
, "יוסי": name
} ()greet: function
;`{this.name}$ אני , שלום` return
{
;{

// יצירת אובייקט שירוש מ-person
;(const student = Object.create(person
; "דני" = student.name
} ()student.study = function
;`{return `${this.name
;{

// "שלום, אני דני" - ירושה מ-person
// "דני לומד" // ;()console.log(student.study
``
```

משתנים מפורט - Detailed Variables

הכרזה על משתנים - כל הדרכים:

```
javascript``
// var - הדרך הישנה (הימנע!)
; "ישן" = var oldVariable
// "ניתן להכריז שוב"; // בעיה! oldVariable

// let - למשתנים שמשתנים
; "יכול להשתנות" = let changeable
; "השתנה" = changeable
// "שגיאה!" // לא ניתן להכריז שוב let changeable

// const - למשתנים קבועים
; "לא יכול להשתנות" = const constant
// "שגיאה!" // לא ניתן לשנות constant

// const עם אובייקטים - התוכן יכול להשתנות!
; { "יוסי": const person = { name
; "דני" = person.name // זה עובד!
; person.age = 25 // זה גם עובד!
; {} = person // זה לא יעבוד!

// const עם מערכים
; [const numbers = [1, 2, 3
; (numbers.push(4 // עובד - משנה את התוכן
; numbers[0] = 10 // עובד - משנה את התוכן
; [] = numbers // לא עובד - משנה את הרפרנט
  ``
```

Hoisting - תופעה מבלבלת: □

```

        javascript``
        :מה שאתה כותב: //
        (!לא שגיאה!) console.log(myVar); // undefined
        !console.log(myLet); // ReferenceError
        !console.log(myConst); // ReferenceError

        ;"var myVar = "Hello
        ;"let myLet = "World
        ;"! = const myConst

        :מה שהמחשב "רואה" //
        var myVar; // undefined
        (Temporal Dead Zone) נגיש לא אבל // ;let myLet //
        (Temporal Dead Zone) נגיש לא אבל // ;const myConst //

```

```

        console.log(myVar); // undefined
        !console.log(myLet); // ReferenceError
        !console.log(myConst); // ReferenceError

        ;"myVar = "Hello
        ;"myLet = "World
        ;"! = myConst
        ``

```

:Temporal Dead Zone ### - אזור מוות זמני:

```

        javascript``
        let/const עם TDZ //
        !console.log(typeof myLet); // ReferenceError
        ;"let myLet = "Hello

        TDZ בפונקציות //
        } ()function example
        !console.log(x); // ReferenceError
        ;let x = 1
        {

        TDZ עם פרמטרים ברירת מחדל //
        } (function defaultParams(a = b, b = 1
        ;return a + b
        {
        defaultParams(); // ReferenceError! b //
        ``

```

:Block Scope vs Function Scope ###

```

        javascript``
        var - Function Scope //
        } ()function varExample
        { (if (true
        ;var x = 1
        {
        !console.log(x); // 1 - נגיש מחוץ לבלוק!
        {

        let/const - Block Scope //
        } ()function letExample
        { (if (true
        ;let y = 1

```

```

;const z = 2
{
!console.log(y); // ReferenceError //
!console.log(z); // ReferenceError //
}

var דוגמה קלאסית - לולאה עם //
} (++for (var i = 0; i < 3; i
3 ,3 ,3 ידפיס // ;(setTimeout(() => console.log(i), 100
{

let פתרון עם //
} (++for (let i = 0; i < 3; i
2 ,1 ,0 ידפיס // ;(setTimeout(() => console.log(i), 100
{
..

```

סוגי נתונים מפורט - Detailed Data Types

Primitive Types - טיפוסים פרימיטיביים:

```

Number #### מספרים
javascript``
מספרים שלמים //
;let integer = 42
;let negative = -17
;let zero = 0

מספרים עשרוניים //
;let decimal = 3.14159
let scientific = 2.5e6; // 2,500,000
let smallScientific = 1.5e-3; // 0.0015

מספרים מיוחדים //
;let infinity = Infinity
;let negativeInfinity = -Infinity
;let notANumber = NaN

בדיקות מספרים //
console.log(Number.isInteger(42)); // true
console.log(Number.isInteger(42.0)); // true
console.log(Number.isInteger(42.1)); // false

console.log(Number.isFinite(42)); // true
console.log(Number.isFinite(Infinity)); // false

console.log(Number.isNaN(NaN)); // true
(NaN לא מספר, אבל לא NaN) console.log(Number.isNaN("hello")); // false

החרות מספרים //
console.log(Number("42")); // 42
console.log(Number("42.5")); // 42.5
console.log(Number("hello")); // NaN
console.log(parseInt("42px")); // 42
console.log(parseFloat("42.5px")); // 42.5

בסיסי מספרים //
let binary = 0b1010; // 10

```

```

        באוקטלי let octal = 0o755; // 493
        בהקסדצימלי let hex = 0xFF; // 255

        BigInt - מספרים גדולים מאוד
        ;let bigNumber = 123456789012345678901234567890n
        ;("let anotherBig = BigInt("123456789012345678901234567890
        ``

        String #####
        javascript``
        דרכים ליצירת מחרוזות
        ;'גרשיים בודדים' = let single
        ;"גרשיים כפולים" = let double
        ;`תבנית מחרוזת` = let template

        Escape characters //
        ;"שורה ראשונה\משורה שנייה\טאב" = let escaped
        ;"שלום\" = let quote
        ;"C:\\Users\\Name :נתיב" = let backslash

        Template literals - תבניות מחרוזות
        ;"יוסי" = let name
        ;let age = 25
        ;`${age}$ בן ,{name}$ שלום` = let message

        מחרוזות מרובות שורות
        ` = let multiline
        זוהי מחרוזת
        מרובת שורות
        עם הזחות
        ;`

        Tagged template literals //
        } (function highlight(strings, ...values
        } <= (return strings.reduce((result, string, i
        `<return result + string + (values[i] ? `<mark>${values[i]}</mark>
        ;('' :
        ;('' ,{
        {

        ;`${age}$ בן ,{name}$ שלום`let highlighted = highlight
        "<mark>25</mark> בן ,<mark/>יוסי<mark> שלום" //

        מתודות מחרוזת חשובות
        ;"let text = "Hello World
        console.log(text.length); // 11
        console.log(text.charAt(0)); // "H
        console.log(text.charCodeAt(0)); // 72
        console.log(text.indexOf("o")); // 4
        console.log(text.lastIndexOf("o")); // 7
        console.log(text.includes("World")); // true
        console.log(text.startsWith("Hello")); // true
        console.log(text.endsWith("World")); // true
        console.log(text.slice(0, 5)); // "Hello
        console.log(text.substring(0, 5)); // "Hello
        (console.log(text.substr(0, 5)); // "Hello" (deprecated
        console.log(text.toLowerCase()); // "hello world
        console.log(text.toUpperCase()); // "HELLO WORLD

```

```

        console.log(text.trim
"console.log(text.replace("World", "JavaScript")); // "Hello JavaScript
["console.log(text.split(" "); // ["Hello", "World
\`\`\`

```

Boolean ##### - בוליאני

javascript``

// ערכים בוליאניים

;let isTrue = true

;let isFalse = false

// החרה לבוליאני

console.log(Boolean(1)); // true

console.log(Boolean(0)); // false

console.log(Boolean("")); // false

console.log(Boolean("hello")); // true

console.log(Boolean(null)); // false

console.log(Boolean(undefined)); // false

console.log(Boolean(NaN)); // false

(!מערך ריק) console.log(Boolean([])); // true

(!אובייקט ריק) console.log(Boolean({})); // true

// קיצור עם !!

console.log(!!"hello"); // true

console.log(!!0); // false

\`\`\`

Truthy □ ו-Falsy - חושג קריטי:

javascript``

// ערכים Falsy (נחשבים false בהקשר בוליאני):

if (false) { } // false

if (0) { } // 0

if (-0) { } // -0

if (0n) { } // BigInt 0

if ("") { } // "" if

if (null) { } // null

if (undefined) { } // undefined

if (NaN) { } // NaN

// כל השאר הם Truthy

{ ;("if (true) { console.log("true

{ ;("מספר חיובי")if (1) { console.log

{ ;("מספר שלילי")if (-1) { console.log

!truthy // { ;("מחרוזת עם 0")if ("0") { console.log

!false"); } // truthy עם "מחרוזת עם false") { console.log

!truthy // { ;("מערך ריק")if ([]) { console.log

!truthy // { ;("אובייקט ריק")if ({}) { console.log

!truthy // { ;("פונקציה")if (function(){}) { console.log

// שימושים מעשיים

} (function greet(name

if (name // בדוקה אם name קיים ולא ריק

;`{name}\$שלוש` return

{

;שלוש אורח" return

{

```

        default values עם //
    } ("אורח" = function greetWithDefault(name
        ;`{name}$ שלום` return
        {

        logical operators עם //
    } (function greetWithOr(name
        ;`{"אורח" || name}$ שלום` return
        {
        ...

Null vs Undefined - הבדל חשוב:

        javascript``
    undefined - משתנה שהוכרז אבל לא קיבל ערך
        ;let undefinedVar
        console.log(undefinedVar); // undefined
    "console.log(typeof undefinedVar); // "undefined

    null - ערך ריק במכוון
        ;let nullVar = null
        console.log(nullVar); // null
    "console.log(typeof nullVar); // "object
        (באג היסטורי!)

    השוואות //
    console.log(null == undefined); // true
    console.log(null === undefined); // false

    בדיקות //
    console.log(null == null); // true
    console.log(undefined == undefined); // true
    console.log(null === null); // true
    console.log(undefined === undefined); // true

    שימושים מעשיים //
    } (function findUser(id
        // אם לא נמצא משתמש, החזר null (לא undefined)
    ;return users.find(user => user.id === id) || null
    {

    בדיקה אם ערך קיים //
    } (if (value !== null
        // הערך קיים
    {

    בדיקה מפורשת // } (if (value !== null && value !== undefined
        // הערך קיים
    {

    (nullish coalescing (ES2020 עם //
    const result = value ?? "ברירת מחדל"; // רק אם null או undefined
    ...

Comprehensive Functions - מקיף פונקציות ##

    דרכים שונות להגדיר פונקציות: ###

    Function Declaration .1 - הכרזת פונקציה ####

```

```

        javascript``
        (hoisted) מוגדרת בכל הקובץ //
    } (function regularFunction(name, age = 0
        ;`{age}$ בן ,{name}$ שלום` return
        {

        : ניטן לקרוא לפני ההגדרה:
    !console.log(regularFunction("יוסי")) ; // עובד!

        עם פרמטרים מרובים //
    } (function sum(a, b, c
        ;return a + b + c
        {

        return מוקדם //
    } (function checkAge(age
        } (if (age < 0
        ;"גיל לא תקין" return
        {
        } (if (age < 18
        ;"קטין" return
        {
        ;"בוגר" return
        {
        ...

```

2. Function Expression - ביטוי פונקציה

```

        javascript``
        לא hoisted - חייב להגדיר לפני השימוש
    } (const functionExpression = function(name
        ;`{name}$ שלום` return
        ;{

```

```

        Named function expression //
    } (const namedExpression = function greet(name
        // השם 'greet' זמין רק בתוך הפונקציה
        } (if (!name
        ;return greet
        {
        ;`{name}$ שלום` return
        ;{

```

```

    IIFE - Immediately Invoked Function Expression //
        } ()function)
        ;("פונקציה שרצה מיד")console.log
        ;()({

```

```

        עם פרמטרים IIFE //
    } (function(message)
        ;(console.log(message
        ;("הודעה מיידית"))({
        ...

```

3. Arrow Functions - פונקציות חץ

```

        javascript``
        // תחביר בסיסי
    } <= (const arrowFunction = (name
        ;`{name}$ שלום` return

```

```

;{

    // תחביר קצר - return מרוחז
    ;`{name}$` שלום` <= const shortArrow = name

    // ללא פרמטרים
    ;"שלום עולם" <= () = const noParams

    // פרמטר יחיד ללא סוגריים
    ;`{name}$` שלום` <= const singleParam = name

    // פרמטרים מרובים
    ;`{age}$` בן ,{name}$` שלום` <= (const multipleParams = (name, age

    // החזרת אובייקט - צריך סוגריים
    ;({`{name}$` שלום` :const returnObject = name => ({ name: name, greeting

    // פונקציה מורכבת
    } <= (const complexArrow = (numbers
    ;(const sum = numbers.reduce((acc, num) => acc + num, 0
    ;const average = sum / numbers.length
    ;{ return { sum, average
    ;{
    ;{

    ### □□ הבדלים חשובים בין סוגי הפונקציות:

    ##### this Binding - קישור הקונטקסט
    javascript`
    } = const person
    , "יוסי": name

    // פונקציה רגילה - this מצביע על person
    } ()regularMethod: function
    "יוסי" // ;(console.log(this.name

    // פונקציה פנימית - this מצביע על window/global
    } ()function innerFunction
    console.log(this.name); // undefined
    {
    ;()innerFunction

    // פתרון עם bind
    ;(const boundFunction = innerFunction.bind(this
    "יוסי" // ;()boundFunction

    // פתרון עם arrow function
    } <= () = const arrowInner
    "יוסי" // ;(console.log(this.name
    ;{
    ;()arrowInner
    ,{

    // Arrow function - this מצביע על הקונטקסט החיצוני
    } <= () :arrowMethod
    (window.name או) console.log(this.name); // undefined
    ,{

```

```

        arrow function שמחזירה //
        } ()createGreeter: function
            } <= () return
"שלום יוסי" // ;({this.name}$`שלום`)console.log
;{
    {
        ;{

"יוסי", "יוסי", undefined, "יוסי" // ;()person.regularMethod
person.arrowMethod(); // undefined

;()const greeter = person.createGreeter
"שלום יוסי" // ;()greeter
``,`

Arguments Object vs Rest Parameters ####
javascript``
(ישן) - arguments או בייקט דמוי מערך //
} ()function oldWay
;(console.log(arguments.length
;([console.log(arguments[0

        המרה למערך //
;(const argsArray = Array.from(arguments
או //
;(const argsArray2 = Array.prototype.slice.call(arguments

!arrow functions עם עובד //
{

    (מודרני) - Rest parameters //
    } (function newWay(...args
    ;(console.log(args.length
    ;([console.log(args[0

        !זה כבר מערך //
;(args.forEach(arg => console.log(arg
;(args.map(arg => arg * 2

        arrow functions עם עובד גם //
{

    } <= (const arrowWithRest = (...args
;(return args.reduce((sum, num) => sum + num, 0
;{

        שילוב עם פרמטרים רגילים //
} (function mixedParams(first, second, ...rest
;(first, "ראשון")console.log
;(second, "שני")console.log
;(rest, "השאר")console.log
{

;(mixedParams(1, 2, 3, 4, 5
1 : ראשון //
2 : שני //
[5, 4, 3] : השאר //
`,``,`

```

```

### פרמטרים מתקדמים:

Default Parameters - פרמטרים ברירת מחדל ####
    javascript``
        בסיסי //
        } (age = 0 , "אורח" = function greet(name
            ;`{age}$ בן , {name}$ שלום` return
            {

                עם ביטויים //
        } (()function createUser(name, timestamp = Date.now
            ;{ return { name, createdAt: timestamp
            {

                עם פונקציות //
        } ()function getDefaultName
            ;"משתמש אנונימי" return
            {

        } (()function register(name = getDefaultName
            ;`{name}$ נרשם` return
            {

                פרמטרים תלויים //
        } (function createRect(width, height = width
            ;{ return { width, height
            {

                עם destructuring //
        } ({}) = { age = 0 , "אורח" = function processUser({ name
            ;`({return `${name} (${age
            {

                "אורח (0)" // ;()processUser
                "יוסי (0)" // ;({ "יוסי":processUser({ name
                "דנה (25)" // ;({ age: 25 , "דנה":processUser({ name
                ``

Destructuring Parameters - פירוק פרמטרים ####
    javascript``
        פירוק אובייקט //
        } ({ function createUser({ name, email, age = 18
            } return
            ,name
            ,email
            ,age
            ()id: Math.random
            ;{
            {

;({ "email: "yossi@example.com , "יוסי":createUser({ name

        פירוק מערך //
        } ([function processCoordinates([x, y, z = 0
            ;{ return { x, y, z
            {

```

```

{ processCoordinates([10, 20]); // { x: 10, y: 20, z: 0

        פירוק מקוון //
    })function processOrder
    ,{ user: { name, email
            ,items
    { "shipping: { address, method = "standard
                    } ({
                        } return
                        ,customerName: name
                        ,customerEmail: email
                        ,itemCount: items.length
                        ,shippingAddress: address
                        shippingMethod: method
                    };{
                        {
                        ...

Higher-Order Functions ###
:פונקציות מסדר גבוה:

        javascript``
        פונקציה שמקבלת פונקציה כפרמטר //
    } (function executeOperation(a, b, operation
        ;(return operation(a, b
            {

                ;const add = (x, y) => x + y
                ;const multiply = (x, y) => x * y

        console.log(executeOperation(5, 3, add)); // 8
        console.log(executeOperation(5, 3, multiply)); // 15

        פונקציה שמחזירה פונקציה //
    } (function createMultiplier(factor
        } (return function(number
        ;return number * factor
            ;{
                {

                ;(const double = createMultiplier(2
                ;(const triple = createMultiplier(3

        console.log(double(5)); // 10
        console.log(triple(5)); // 15

        פונקציה שמחזירה פונקציות - Currying //
        } (function curry(fn
        } (return function curried(...args
        } (if (args.length >= fn.length
        ;(return fn.apply(this, args
            } else {
                } (return function(...args2
        ;((return curried.apply(this, args.concat(args2
            ;{
                {
                ;{
                {

```

```

;const add3 = (a, b, c) => a + b + c
; (const curriedAdd = curry(add3

console.log(curriedAdd(1)(2)(3)); // 6
console.log(curriedAdd(1, 2)(3)); // 6
console.log(curriedAdd(1)(2, 3)); // 6

Partial Application //
} (function partial(fn, ...presetArgs
} (return function(...laterArgs
;(return fn(...presetArgs, ...laterArgs
;{
{

;(const add10 = partial(add, 10
console.log(add10(5)); // 15
```

Function Methods ###
:חנותות פונקציה:

javascript``
} = const person
, "יוטי": name
} (greet: function(greeting, punctuation
;`{return `${greeting} ${this.name}${punctuation
{
;{

call // - קריאה מיידית עם this מוגדר
;(("!" , "שלום" , { "דנה" : console.log(person.greet.call({ name
"שלום דנה" //

apply // - כמו call אבל עם מערך פרמטרים
;([["?" , "היי"] , { "מיכל" : console.log(person.greet.apply({ name
"היי מיכל?" //

bind // - יצירת פונקציה חדשה עם this קבוע
;({ "דנה" : const greetDana = person.greet.bind({ name
"שלום דנה" // ;(("!" , "שלום") console.log(greetDana

bind // עם פרמטרים חלקיים
;("שלום" , { "דנה" : const greetDanaHello = person.greet.bind({ name
"שלום דנה" // ;(("!") console.log(greetDanaHello

event handlers - שימוש מעשי //
} class Button
} (constructor(element
;this.element = element
;this.clickCount = 0

this כדי לשחזר על bind //
this.element.addEventListener('click',
;((this.handleClick.bind(this
{

} ()handleClick
; ++this.clickCount
;(`{this.clickCount}$ןלחץ) console.log

```

```
{
 ...
}
```

Comprehensive Objects - אובייקטים מקיף ##

### יצירת אובייקטים - כל הדרכים:

Object Literal .1 #### - הדרך הפשוטה

```
javascript``
// בסיסי
} = const person
, "יוסי": name
, age: 25
"תל אביב": city
};

// עם חתודות
} = const calculator
, result: 0

} (add: function(number
; this.result += number
return this; // method chaining
, {

} (multiply: function(number
; this.result *= number
; return this
, {

} ()get value
; return this.result
, {

reset() { // shorthand method
; this.result = 0
; return this
{
};

; (calculator.add(5).multiply(2).add(3
console.log(calculator.value); // 13

Computed property names //
; "const propertyName = "dynamicProperty
} = const obj
, "ערך דינמי": [propertyName]
"ערך נוסף": [`propertyName}2}$`]
};

Property shorthand //
; "יוסי" = const name
; const age = 25
{ name: name, age: age }-ל // ; { const user = { name, age
````
```

Constructor Functions .2 #### - פונקציות בנאי

```

        javascript``
        } (function Person(name, age
            ;this.name = name
            ;this.age = age

(instance לא מומלץ - יוצרת עותק לכל instance
        } ()this.greet = function
        ;`{this.name}$ שלום` return
        ;{
        {

        (מומלץ) prototype-ב
        } ()Person.prototype.introduce = function
        ;`{this.age}$ בן ,{this.name}$ אני` return
        ;{

        } ()Person.prototype.haveBirthday = function
            ;++this.age
        ;`{this.age}$ עכשיו בן ! יום הולדת שמח!` return
        ;{

        instances יצירת //
        ;(25 , "יוסי")const person1 = new Person
        ;(30 , "דנה")const person2 = new Person

        "25 בן יוסי , אני" // ;(()console.log(person1.introduce
        "31 בן עכשיו ! יום הולדת שמח!` return
        ;`{this.name}$ שלום` return
        ;{
        ;(const person = Object.create(personPrototype
            ;(25 , "יוסי")person.init

        // או בשורה אחת
        ;(30 , "דנה")const person2 = Object.create(personPrototype).init

        // עם properties מוגדרות
        } ,const person3 = Object.create(personPrototype
            :name
            , "מיכל":value
            ,writable: true
            ,enumerable: true
            configurable: true
            ,{
            } :age

```

###3 .Object.create() - יצירה עם prototype מוגדר

```

        javascript``
        } = const personPrototype
        } (init: function(name, age
            ;this.name = name
            ;this.age = age
            ;return this
            ,{

            } ()greet: function
            ;`{this.name}$ שלום` return
            {
            ;{

        ;(const person = Object.create(personPrototype
            ;(25 , "יוסי")person.init

        // או בשורה אחת
        ;(30 , "דנה")const person2 = Object.create(personPrototype).init

        // עם properties מוגדרות
        } ,const person3 = Object.create(personPrototype
            :name
            , "מיכל":value
            ,writable: true
            ,enumerable: true
            configurable: true
            ,{
            } :age

```

```

        ,value: 28
        ,writable: true
        ,enumerable: true
        configurable: true
    }
    ;({
    ...

(+ES6) החודרני - Classes .4 ####
    javascript``
    } class Person
    Constructor //
    } (constructor(name, age
    ;this.name = name
    ;this.age = age
this._id = Math.random(); // private-like property
    {

        Method //
        } ()greet
    ;`{this.name}$ שולח` return
    {

        Static method //
        } ()static species
    ;"return "Homo sapiens
    {

        Getter //
        } ()get info
    ;`{this.age}$ בן ,{return `${this.name
    {

        Setter //
        } (set age(newAge
    } (if (newAge >= 0 && newAge <= 150
    ;this._age = newAge
        } else {
    ;("גיל לא תקין")throw new Error
        {
        {

        } ()get age
    ;return this._age
    {

        (Private method (ES2022 //
        } ()calculateSomething#
    ;return this._id * 100
    {

    Public method that uses private method //
        } ()getCalculation
    ;()return this.#calculateSomething
    {
    {

    ;(25 , "יוסי")const person = new Person

```

```

"שלום יוסי" // ;(()console.log(person.greet
"console.log(Person.species()); // "Homo sapiens
"25 יוסי, בן" // ;(console.log(person.info
`

```

גישה לתכונות - Property Access

```

    javascript``
    } = const person
    , "יוסי": name
    , age: 25
    , "full-name": "יוסי כהן"
    , "מספר כמפתח": "123"
    ;{

    Dot notation //
    "יוסי" // ;(console.log(person.name
    console.log(person.age); // 25

    Bracket notation //
    "יוסי" // ;(["console.log(person["name
    "יוסי כהן" - חובה עם מקף // ;(["console.log(person["full-name
    "מספר כמפתח" // ;(["console.log(person["123

    גישה דינמית //
    ;"const property = "age
    console.log(person[property]); // 25

    עם משתנים //
    ;"const prefix = "full
    "יוסי כהן" // ;(["console.log(person[prefix + "-name

    (Optional chaining (ES2020 //
    } = const user
    } :profile
    , "יוסי": name
    } :address
    "תל אביב": city
    {
        {
    ;{

    "יוסי" // ;(console.log(user?.profile?.name
    "תל אביב" // ;(console.log(user?.profile?.address?.city
    console.log(user?.profile?.phone?.number); // undefined (!לא שגיאה)

    עם מתודות //
    // לא יקרא לפונקציה אם לא קיימת ;()

    עם מערכים //
    ] = const users
    , { "יוסי": name }
    , null
    { "דנה": name }
    ;[

    "יוסי" // ;(console.log(users?.[0]?.name
    console.log(users?.[1]?.name); // undefined

```

```
console.log(users?.[10]?.name); // undefined
`
```

הוספה, שינוי ומחיקה של תכונות:

```
javascript``
;{ "יוסי" :const person = { name

// הוספת תכונות
;person.age = 25
; "תל אביב" = ["person["city
; "יוסי כהן" = ["person["full-name

// שינוי תכונות
;person.age = 26
; "חיפה" = ["person["city

// מחיקת תכונות
;delete person.age
;["delete person["full-name

// בדיקת קיום תכונות
console.log("name" in person); // true
console.log("age" in person); // false
console.log(person.hasOwnProperty("name")); // true
console.log(person.hasOwnProperty("toString")); // false (ירושה)

// Object.hasOwnProperty() - חדש יותר
console.log(Object.hasOwnProperty(person, "name")); // true

// בדיקה אם תכונה undefined
console.log(person.age === undefined); // true
console.log(person.nonExistent === undefined); // true - לא מבדיל!

// בדיקה נכונה
} (if ("age" in person
;("console.log("age
} else {
;("console.log("age
{
`
```

העתקת אובייקטים - בעיה נפוצה:

```
javascript``
} = const original
, "יוסי" :name
, ["קריאה", "ספורט"] :hobbies
} :address
, "תל אביב" :city
"1 הרצל" :street
{
;{

// העתקה שטחית - רק הרמה הראשונה
;{ const shallowCopy1 = { ...original
;(const shallowCopy2 = Object.assign({}, original
```

```

// בעיה: שינוי במערך או אובייקט פנימי משפיע על המקור!
;("בישול")shallowCopy1.hobbies.push
; "חיפה" = shallowCopy1.address.city

!השתנה - ["בישול", "ספורט", "קריאה"] // ;(console.log(original.hobbies
!השתנה - "חיפה" // ;(console.log(original.address.city

// העתקה עמוקה - כל הרמות
;(const deepCopy1 = JSON.parse(JSON.stringify(original
Date, RegExp, undefined, Symbol , פונקציות, לא עובד עם
// העתקה עמוקה עם structuredClone (חדש)
;(const deepCopy2 = structuredClone(original

// העתקה עמוקה ידנית
} (function deepClone(obj
} ("if (obj === null || typeof obj !== "object
;return obj
{
} (if (obj instanceof Date
;(()return new Date(obj.getTime
{
} (if (obj instanceof Array
;((return obj.map(item => deepClone(item
{
} ("if (typeof obj === "object
;{} = const cloned
} (for (let key in obj
} ((if (obj.hasOwnProperty(key
;([cloned[key] = deepClone(obj[key
{
{
;return cloned
{
{
;(const deepCopy3 = deepClone(original
``,`

Object Methods ### - חתודות אובייקט:

    javascript``
    } = const person
    , "יוסי": name
    , age: 25
    "תל אביב": city
    ;{

// Object.keys() - מערך של מפתחות
["const keys = Object.keys(person); // ["name", "age", "city

// Object.values() - מערך של ערכים
["const values = Object.values(person
["תל אביב", 25, "יוסי"] //

// Object.entries() - מערך של [מפתח, ערך]

```

```

        ;(const entries = Object.entries(person
[[["תל אביב", "age", 25], ["city"], ["יוטי", "name"]]] //

        entries - מ-Object.fromEntries //
        ;(const newObj = Object.fromEntries(entries

        Object.assign //
        ;{ "const defaults = { theme: "light", language: "he
        ;{ "const userPrefs = { theme: "dark
;(const settings = Object.assign({}, defaults, userPrefs
        { "theme: "dark", language: "he } //

        spread operator (מורחב יותר) עם //
        ;{ const settings2 = { ...defaults, ...userPrefs

        Object.freeze //
        ;({ "const frozenObj = Object.freeze({ name
strict fail) או שגיאה ב-silent (mode

        Object.seal //
        ;({ age: 25, "const sealedObj = Object.seal({ name
        sealedObj.name = "דנה"; // עובר - שינוי תכונות קיימות
        sealedObj.city = "חיפה"; // לא עובר - הוספת תכונות חדשות
        delete sealedObj.age; // לא עובר - מחיקת תכונות

        Object.preventExtensions //
        ;({ "const nonExtensible = Object.preventExtensions({ name
        nonExtensible.name = "דנה"; // עובר
        nonExtensible.age = 25; // לא עובר

        בדיקות מצב //
        console.log(Object.isFrozen(frozenObj)); // true
        console.log(Object.isSealed(sealedObj)); // true
        console.log(Object.isExtensible(nonExtensible)); // false
        ``

```

זהו חלק נרחב מהמדריך המתקדם ל-JavaScript. האם תרצה שאמשיך עם החלקים הבאים (מערכים, Promises, DOM, וכו')?

מערכים מתקדמים - Advanced Arrays

יצירת מערכים - כל הדרכים:

```

javascript``
// דרכים ליצירת מערכים
const arr1 = [1, 2, 3]; // literal notation
const arr2 = new Array(1, 2, 3); // constructor
;(const arr3 = new Array(5 // מערך עם 5 מקומות ריקים
;(const arr4 = Array.of(1, 2, 3 // תחיד יוצר מערך עם הערכים
["const arr5 = Array.from("hello"); // ["h", "e", "l", "l", "o
[const arr6 = Array.from({length: 3}, (_, i) => i); // [0, 1, 2

// Typed Arrays - מערכים מטיפוס ספציפי
const int8Array = new Int8Array([1, 2, 3]); // 8-bit integers
const float32Array = new Float32Array([1.1, 2.2, 3.3]); // 32-bit floats
const uint8Array = new Uint8Array(10); // 10 unsigned 8-bit integers

```

```

// מערכים מקוננים
] = const matrix
  , [3, 2, 1]
  , [6, 5, 4]
  , [9, 8, 7]
;

// מערך מעורב
; [[3, 2, 1], { "יוסי": const mixed = [1, "hello", true, null, {name

### מתודות מערך מתקדמות:

#### מתודות שמשנות את המערך (Mutating)
javascript`
; ["תפוח", "בננה", "תפוז"] = const fruits

// הוספה והסרה
fruits.push("אבטיח"); // הוספה לסוף
fruits.unshift("תות"); // הוספה להתחלה
const lastFruit = fruits.pop(); // הסרה מהסוף
const firstFruit = fruits.shift(); // הסרה מההתחלה

// splice - הסרה והוספה במקום ספציפי
const removed = fruits.splice(1, 2, "מנגו", "פפאיה");
// מהאינדקס 1, הסר 2 פריטים, הוסף "מנגו" ו"פפאיה"

// מיון
; const numbers = [3, 1, 4, 1, 5, 9]
numbers.sort(); // מיון לקסיקוגרפי: [1, 1, 3, 4, 5, 9]
// מיון מספרי עולה (numbers.sort((a, b) => a - b))
// מיון מספרי יורד (numbers.sort((a, b) => b - a))

// הפיכה
numbers.reverse(); // הופך את סדר האלמנטים

// מילוי
const zeros = new Array(5).fill(0); // [0, 0, 0, 0, 0]
; const pattern = [1, 2, 3]
pattern.fill("א", 1, 2); // [1, "א", 3] - מילוי מאינדקס 1 עד 2

// copyWithin - העתקה פנימית
; const arr = [1, 2, 3, 4, 5]
arr.copyWithin(0, 3, 4); // [4, 2, 3, 4, 5] - העתק מאינדקס 3-4 לאינדקס 0

#### מתודות שלא משנות את המערך (Non-mutating)
javascript`
; const numbers = [1, 2, 3, 4, 5]

// map - יצירת מערך חדש עם שינוי
const doubled = numbers.map(num => num * 2); // [2, 4, 6, 8, 10]
const strings = numbers.map(num => `${num} מספר`);

// filter - סינון
const evens = numbers.filter(num => num % 2 === 0); // [2, 4]
const odds = numbers.filter(num => num % 2 === 1); // [1, 3, 5]

```

```

        צמצום לערך יחיד - reduce //
const sum = numbers.reduce((acc, num) => acc + num, 0); // 15
const product = numbers.reduce((acc, num) => acc * num, 1); // 120

        צמצום מימין לשמאל - reduceRight //
;(const rightSum = numbers.reduceRight((acc, num) => acc + num, 0

        מציאת אלמנט ראשון - find //
const found = numbers.find(num => num > 3); // 4
const foundIndex = numbers.findIndex(num => num > 3); // 3

        בדיקה אם לפחות אחד מתאים - some //
const hasEven = numbers.some(num => num % 2 === 0); // true

        בדיקה אם כולם מתאימים - every //
const allPositive = numbers.every(num => num > 0); // true

        בדיקת קיום - includes //
const hasThree = numbers.includes(3); // true

        מציאת אינדקס - indexOf / lastIndexOf //
const index = numbers.indexOf(3); // 2
const lastIndex = numbers.lastIndexOf(3); // 2

        חיתוך (לא משנה את המקור) - slice //
[const sliced = numbers.slice(1, 4); // [2, 3, 4
[const fromIndex = numbers.slice(2); // [3, 4, 5
[const lastTwo = numbers.slice(-2); // [4, 5

        חיבור מערכים - concat //
;[const moreNumbers = [6, 7, 8
[const combined = numbers.concat(moreNumbers); // [1, 2, 3, 4, 5, 6, 7, 8

        הפיכה למחרוזת - join //
"const joined = numbers.join(", "); // "1, 2, 3, 4, 5
"const noSeparator = numbers.join(""); // "12345
``,`

### מתודות מערך מודרניות (+ES6)

        javascript``
;[const numbers = [1, 2, 3, 4, 5

        mapping עם Array.from //
[const doubled = Array.from(numbers, x => x * 2); // [2, 4, 6, 8, 10

        flatMap - map + flat //
;[[const nested = [[1, 2], [3, 4], [5
const flattened = nested.flatMap(arr => arr.map(x => x * 2)); // [2, 4,
[6, 8, 10

        פיתוח מערכים מקוננים - flat //
;[[[const deepNested = [1, [2, 3], [4, [5, 6
[[const flat1 = deepNested.flat(); // [1, 2, 3, 4, [5, 6
[const flat2 = deepNested.flat(2); // [1, 2, 3, 4, 5, 6
פיתוח חלל // ;(const flatAll = deepNested.flat(Infinity

        entries, keys, values //

```

```

    } (()for (const [index, value] of numbers.entries
      ;(`{console.log(`${index}: ${value
        {

      } (()for (const index of numbers.keys
      ;(`{index}$ :אינדקס`)console.log
        {

      } (()for (const value of numbers.values
      ;(`{value}$ :ערך`)console.log
        {

      (ES2022) - גישה עם אינדקסים שליליים //
      console.log(numbers.at(-1)); // 5
      console.log(numbers.at(-2)); // 4
      ``

      ### דוגמאות מעשיות למערכים:

      #### עיבוד נתונים
      javascript``
      ] = const users
      ,{ salary: 15000 , "תל אביב" :age: 25, city , "יוטי" :name }
      ,{ salary: 18000 , "חיפה" :age: 30, city , "דנה" :name }
      ,{ salary: 16000 , "תל אביב" :age: 28, city , "מיכל" :name }
      { salary: 20000 , "ירושלים" :age: 35, city , "אבי" :name }

      ;[

      // סינון וחישוב
      ;("תל אביב" === const tlvUsers = users.filter(user => user.city
      ;(const highEarners = users.filter(user => user.salary > 16000
      const averageAge = users.reduce((sum, user) => sum + user.age, 0) /
      ;users.length

      // קיבוץ לפי עיר
      } <= (const usersByCity = users.reduce((acc, user
        } ([if (!acc[user.city
          ;[] = [acc[user.city
            {
          ;(acc[user.city].push(user
            ;return acc
          ;({} ,{

      // מיון מורכב
      } <= (const sortedUsers = users.sort((a, b
        // מיון לפי עיר ואז לפי שכר
        } (if (a.city !== b.city
      ;(return a.city.localeCompare(b.city
        {
      ;return b.salary - a.salary
      ;({

      // שרשור פטולות
      const result = users
      (filter(user => user.age > 25.
      (({ map(user => ({ ...user, bonus: user.salary * 0.1.
      (sort((a, b) => b.salary - a.salary.
      ;(slice(0, 2.

```

```

...

    API עבודה עם ###
    javascript``
    API עיבוד תגובות //
    } ()async function processApiData
    ])const responses = await Promise.all
        ,('fetch('/api/users
        ,('fetch('/api/posts
        ('fetch('/api/comments
        ;({

    )const [users, posts, comments] = await Promise.all
        ((()responses.map(response => response.json
        ;(

        חיבור נתונים //
    }) <= const enrichedPosts = posts.map(post
        ,post...
    , (author: users.find(user => user.id === post.userId
    commentCount: comments.filter(comment => comment.postId ===
        post.id).length
        ;(({

    ;return enrichedPosts
    {
    ...

Promises ו-Async/Await מקיף ##

## חבוא ל-Promises:
Promise הוא אובייקט שמייצג השלמה או כישלון עתידי של פעולה אסינכרונית.

    javascript``
    // יצירת Promise בסיסי
    } <= (const myPromise = new Promise((resolve, reject
        ;const success = Math.random() > 0.5

        } <= ()setTimeout
        } (if (success
        ;("הפעולה הצליחה!")resolve
        } else {
        ;(("הפעולה נכשלה!")reject(new Error
            {
            ;(1000 ,{
            ;({

    // שימוש ב-Promise
    myPromise
    } <= then(result.
        ;(result , "הצלחה")console.log
    return "תוצאה נוספת" ; // מועבר ל-then הבא
    ({
        } <= then(result.
        ;(result , "שלב שני")console.log
    ({
        } <= catch(error.
        ;(error.message , "שגיאה")console.error

```

```

        } <= ())finally.
        ;("תחיד רץ בסוף")console.log
        ;({
            ...

            :Promise חצבי ###
            javascript``
            מחתיך - Pending //
            {<fulfilled}const pendingPromise = new Promise

            הושלם בהצלחה - Fulfilled //
            ;("הצלחה")const fulfilledPromise = Promise.resolve

            נכשל - Rejected //
            ;(("כישלון")const rejectedPromise = Promise.reject(new Error

            בדיקת מצב (לא ישיר - רק לדוגמה) //
            {<fulfilled}console.log(fulfilledPromise); // Promise {<fulfilled
            ``

            :Promise מתודות - Promise Methods ###
            javascript``
            Promise.all - מחכה לכל ה-Promises //
            ;(const promise1 = Promise.resolve(1
            ;(const promise2 = Promise.resolve(2
            ;(const promise3 = Promise.resolve(3

            ([Promise.all([promise1, promise2, promise3
            } <= then(results.
            [console.log(results); // [1, 2, 3
            ({
            } <= catch(error.
            // אם אחד נכשל, הכל נכשל
            ;(error, "Promises-מה נכשל:")console.error
            ;({

            Promise.allSettled - מחכה לכולם, גם אם נכשלים //
            ([("שגיאה")Promise.allSettled([promise1, promise2, Promise.reject
            } <= then(results.
            } <= (results.forEach((result, index
            } ('if (result.status === 'fulfilled
            ;(result.value, `הצליח: ${index} {console.log(`Promise ${index
            } else {
            ;(result.reason, `נכשל: ${index} {console.log(`Promise ${index
            {
            ;({
            ;({

            Promise.race - הראשון שמסתיים //
            ])Promise.race
            ,((100, ("מהיר")new Promise(resolve => setTimeout(() => resolve
            ((200, ("איטי")new Promise(resolve => setTimeout(() => resolve
            ([
            } <= then(result.
            "מהיר" // ;(console.log(result
            ;({

```

```

        Promise.any //
        ])Promise.any
        , ("שגיאה 1")Promise.reject
        , ("הצלחה")Promise.resolve
        ("שגיאה 2")Promise.reject
        ([
        ] <= then(result.
        "הצלחה" // ;(console.log(result
        ({
        } <= catch(error.
        // רק אם כולם נכשלים
        ;(error , "נכשלו: Promises-ה")console.error
        ;({
        ``

        : Async/Await - תחביר מודרני: ###

        javascript``
        // פונקציה אסינכרונית בטיסית
        } (async function fetchUserData(userId
        } try
        ;(`const response = await fetch(`/api/users/${userId

        } (if (!response.ok
        ;(`throw new Error(`HTTP error! status: ${response.status

        {

        ;()const userData = await response.json
        ;return userData
        } (catch (error {
        ;(error , "שגיאה בטעינת נתוני משתמש:")console.error
        // ;throw error הלאה
        {
        {

        // שימוש בפונקציה אסינכרונית
        } (async function displayUser(userId
        } try
        ;(const user = await fetchUserData(userId
        ;(!{user.name}$שלום)console.log
        } (catch (error {
        ;("לא ניתן לטעון את המשתמש")console.log
        {
        {

        // פונקציות אסינכרוניות מקבילות
        } ()async function fetchMultipleUsers
        } try
        // רצות במקביל
        ])const [user1, user2, user3] = await Promise.all
        , (fetchUserData(1
        , (fetchUserData(2
        (fetchUserData(3
        ;([

        ;[return [user1, user2, user3
        } (catch (error {

```

```

;(error , "שגיאה בטעינת חשתחשים")console.error
{
{
async/await עם לולאות //
} (async function processUsers(userIds
;[] = const results

(אחד אחרי השני) //
} (for (const id of userIds
} try
;(const user = await fetchUserData(id
;(results.push(user
} (catch (error {
;(id):`, error}$ חשתחש עם שגיאה`)console.error
{
{

;return results
{

(רצות במקביל) map עם //
} (async function processUsersParallel(userIds
;((const promises = userIds.map(id => fetchUserData(id

} try
;(const results = await Promise.allSettled(promises
return results
('filter(result => result.status === 'fulfilled.
;(map(result => result.value.
} (catch (error {
;(error , "שגיאה כללית")console.error
{
...

### דוגמאות מעשיות לאסינכרוניות

#### טעינת נתונים מ-API
javascript``
} class ApiService
} (constructor(baseUrl
;this.baseUrl = baseUrl
{

} ({} = async request(endpoint, options
;`${const url = `${this.baseUrl}${endpoint
} = const config
} :headers
,'Content-Type': 'application/json'
options.headers...
,{
options...
;{

} try
;(const response = await fetch(url, config

```

```

        } (if (!response.ok
throw new Error(`HTTP ${response.status}:
        ;(`${response.statusText
        {

        ;()return await response.json
        } (catch (error {
;(console.error(`API Error for ${endpoint}:`, error
        ;throw error
        {
        {

        } (async get(endpoint
;(return this.request(endpoint
        {

        } (async post(endpoint, data
} ,return this.request(endpoint
        , 'method: 'POST
        (body: JSON.stringify(data
        ;({
        {

        } (async put(endpoint, data
} ,return this.request(endpoint
        , 'method: 'PUT
        (body: JSON.stringify(data
        ;({
        {

        } (async delete(endpoint
} ,return this.request(endpoint
        'method: 'DELETE
        ;({
        {
        {

        שיחוש //
;(const api = new ApiService('https://api.example.com

        } ()async function manageUser
        } try
        יצירת משתמש //
} , 'const newUser = await api.post('/users
        , 'יוסי כהן': name
        'email: 'yossi@example.com
        ;({

        עדכון משתמש //
} , `{const updatedUser = await api.put(`/users/${newUser.id
        ,newUser...
        age: 25
        ;({

        קבלת כל המשתמשים //
;(const allUsers = await api.get('/users

;(allUsers , 'כל המשתמשים')console.log

```

```

        } (catch (error {
;(error , 'שגיאה בניהול חשתחש:')console.error
        {
            {
                (File API) עם קבצים ####
                javascript``
            } class FileProcessor
            } (static async readFile(file
        } <= (return new Promise((resolve, reject
            ;()const reader = new FileReader

;(reader.onload = event => resolve(event.target.result
;(reader.onerror = error => reject(error

;(reader.readAsText(file
;({
{

        } (static async processImage(file
    } <= (return new Promise((resolve, reject
        ;()const reader = new FileReader

        } <= reader.onload = event
        ;()const img = new Image
        } <= () = img.onload
;('const canvas = document.createElement('canvas
;('const ctx = canvas.getContext('2d

// שינוי גודל לתמונה קטנה יותר
;const maxWidth = 800
;const maxHeight = 600

;let { width, height } = img

    } (if (width > maxWidth
;height = (height * maxWidth) / width
    ;width = maxWidth
    {

    } (if (height > maxHeight
;width = (width * maxHeight) / height
    ;height = maxHeight
    {

        ;canvas.width = width
        ;canvas.height = height

;(ctx.drawImage(img, 0, 0, width, height
;(canvas.toBlob(resolve, 'image/jpeg', 0.8
;{

        ;img.onerror = reject
;img.src = event.target.result
;{

```

```

;reader.onerror = reject
;(reader.readAsDataURL(file
;({
{
} (static async uploadFile(file, url
;()const formData = new FormData
;(formData.append('file', file
} try
},const response = await fetch(url
,'method: 'POST'
body: formData
;({
} (if (!response.ok
;(`throw new Error(`Upload failed: ${response.statusText
{
;()return await response.json
} (catch (error {
;(console.error('Upload error:', error
;throw error
{
{
{
שיחון //
} (async function handleFileUpload(fileInput
;(const files = Array.from(fileInput.files
} (for (const file of files
} try
;(`{file.name}$ : קובץ חסר)console.log
} (('if (file.type.startsWith('image
const processedImage = await
;(FileProcessor.processImage(file
const uploadResult = await
;('FileProcessor.uploadFile(processedImage, '/api/upload
;(uploadResult ,':תמונה הועלתה')console.log
} else {
const uploadResult = await FileProcessor.uploadFile(file,
;('/api/upload
;(uploadResult ,':קובץ הועלה')console.log
{
} (catch (error {
;(file.name):`, error)$ עם קובץ שגוי)console.error
{
{
...
Event Loop ו-Concurrency מפורט ##
:Event Loop הבנת ##
javascript`

```

```

// סדר הביצוע המבלבל
console.log('1 - סינכרוני' // ראשון

// ;(setTimeout(() => console.log('2 - setTimeout (Macro task)'), 0
// ;(('Promise.resolve().then(() => console.log('3 - Promise (Micro task
שלישי

console.log('4 - סינכרוני' // שני

// התוצאה: 1, 4, 3, 2
// Event Loop מעדיף Micro tasks על פני Macro tasks

// דוגמה מורכבת יותר
console.log('התחלה')

;(setTimeout(() => console.log('setTimeout 1'), 0

    ()Promise.resolve
    } <= ()))then.
    ;('console.log('Promise 1
    ;()return Promise.resolve
    ({
    ;(('then(() => console.log('Promise 2.

;(setTimeout(() => console.log('setTimeout 2'), 0

    } <= ()))Promise.resolve().then
    ;('console.log('Promise 3
;(Promise'), 0 setTimeout(() => console.log('setTimeout

;({

;('סוף')console.log

: התוצאה //
: התחלה //
: סוף //
Promise 1 //
Promise 3 //
Promise 2 //
setTimeout 1 //
setTimeout 2 //
Promise בתוך setTimeout //
...

:Microtasks vs Macrotasks ###
javascript``
: (עדיפות גבוהה) Microtasks //
Promise.then/catch/finally - //
()queueMicrotask - //
MutationObserver - //

: (עדיפות נמוכה) Macrotasks //
setTimeout/setInterval - //
(setImmediate (Node.js - //
I/O operations - //
UI rendering - //

```

```

        // דוגמה לשליטה בסדר
    } ()function demonstrateTaskOrder
        ;('console.log('1

        Macrotask //
    ;(setTimeout(() => console.log('2 - setTimeout'), 0

        Microtask //
    ;(('Promise.resolve().then(() => console.log('3 - Promise

        Microtask חידי //
    ;(('queueMicrotask(() => console.log('4 - queueMicrotask

        ;('console.log('5
    {

        ;()demonstrateTaskOrder
        // תוצאה: 2 ,4 ,3 ,5 ,1
        ``

DOM Manipulation ##
    ## בחירת אלמנטים מתקדמת:
        javascript``
        // בוררים מתקדמים
        } = const elements
        // בחירה בסיסית
        ,('byId: document.getElementById('myId
        byClass: document.getElementsByClassName('myClass'), //
        HTMLCollection
        byTag: document.getElementsByTagName('div'), // HTMLCollection

        (מוחלק) Query selectors //
        ,('single: document.querySelector('.class #id
        multiple: document.querySelectorAll('div.class'), // NodeList

        // בוררים מתקדמים
        ,('["attributes: document.querySelectorAll('[data-role="button
        ,('pseudoClasses: document.querySelectorAll('li:nth-child(odd
        ,('combinators: document.querySelectorAll('nav > ul > li

        // בחירה יחסית
        // עולה במעלה העץ // ,('closest: element.closest('.parent-class
        // בדיקה אם תואם לבורר // ,('matches: element.matches('.some-class

        // ילדים ואחים
        // רק אלמנטים // ,children: element.children
        // כולל text nodes // ,childNodes: element.childNodes
        ,firstElementChild: element.firstElementChild
        ,lastElementChild: element.lastElementChild
        ,nextElementSibling: element.nextElementSibling
        ,previousElementSibling: element.previousElementSibling
        parentElement: element.parentElement
    };

        // החרה ל-Array
    ;(('const elementsArray = Array.from(document.querySelectorAll('div
    ;(('const elementsSpread = [...document.querySelectorAll('div

```

```

    ...

    ### יצירה ושינוי של אלמנטים:
    javascript``
    // יצירת אלמנטים
} ([] = function createElement(tag, attributes = {}, children
    ;(const element = document.createElement(tag

        // הוספת תכונות
    } <= ([Object.entries(attributes).forEach(([key, value
        } ('if (key === 'className
        ;element.className = value
    } ('else if (key === 'textContent {
        ;element.textContent = value
    } ('else if (key === 'innerHTML {
        ;element.innerHTML = value
    } (('else if (key.startsWith('data {
        ;(element.setAttribute(key, value
        } else {
        ;element[key] = value
        {
        ;({

        // הוספת ילדים
    } <= children.forEach(child
    } ('if (typeof child === 'string
;((element.appendChild(document.createTextNode(child
    } else {
    ;(element.appendChild(child
    {
    ;({

    ;return element
    {

    // שימוש
    } , 'const button = createElement('button
    , 'className: 'btn btn-primary
    , 'לחץ כאן': textContent
    , 'data-action': 'submit'
    ;({

    ] , { 'const card = createElement('div', { className: 'card
    , ({ 'כותרת': createElement('h3', { textContent
    , ({ 'תוכן הכרטיס': createElement('p', { textContent
    button
    ;([

    // הוספה ל-DOM
    ;(document.body.appendChild(card

    // שינוי תוכן
    } <= (const updateElement = (element, updates
    } <= ([Object.entries(updates).forEach(([key, value
        } ('if (key === 'text
        ;element.textContent = value
        } ('else if (key === 'html {
        ;element.innerHTML = value

```

```

        } ('else if (key === 'classes' {
element.className = Array.isArray(value) ? value.join(' ') :
;value
        } ('else if (key === 'style' {
; (Object.assign(element.style, value
        } ('else if (key === 'attributes' {
} <= ([Object.entries(value).forEach(([attr, val
; (element.setAttribute(attr, val
; ({
{
; ({
; {
//
} ,updateElement(button
, 'טקסט חדש' :text
, ['classes: ['btn', 'btn-success
, { 'style: { backgroundColor: 'green', color: 'white
{ 'attributes: { 'data-updated': 'true
; ({
``

:חלקי Event Handling ###
javascript``
} class EventManager
} ()constructor
; ()this.listeners = new Map
{

// הוספת event listener עם אפשרויות מתקדמות
} ({}) = on(element, event, handler, options
} <= (const wrappedHandler = (e
Pre-processing //
; ()if (options.preventDefault) e.preventDefault
; ()if (options.stopPropagation) e.stopPropagation

Throttling //
} (if (options.throttle
} (if (!handler._throttled
; handler._throttled = true
} <= ())setTimeout
; handler._throttled = false
; (options.throttle , {
; (handler(e
{
; return
{

Debouncing //
} (if (options.debounce
; (clearTimeout(handler._debounceTimer
} <= ())handler._debounceTimer = setTimeout
; (handler(e
; (options.debounce , {
; return
{

; (handler(e

```

```

;{

;(element.addEventListener(event, wrappedHandler, options

    שמירה לניקוי מאוחר יותר //
    ;`{const key = `${element}_${event}
      } ((if (!this.listeners.has(key)
        ;([], this.listeners.set(key
          {
;({ this.listeners.get(key).push({ handler, wrappedHandler

;return this
{

    event listeners הסרת //
    } (off(element, event, handler
    ;`{const key = `${element}_${event}
    ;(const listeners = this.listeners.get(key

    } (if (listeners
const index = listeners.findIndex(l => l.handler ===

; (handler
    } (if (index !== -1
    ;[const { wrappedHandler } = listeners[index]
;(element.removeEventListener(event, wrappedHandler
    ;(listeners.splice(index, 1
    {
    {

;return this
{

    Event delegation //
    } (delegate(container, selector, event, handler
    } <= (container.addEventListener(event, (e
    ;(const target = e.target.closest(selector
    } ((if (target && container.contains(target)
    ;(handler.call(target, e
    {
    ;({

;return this
{

    Custom events //
    } ({} = emit(element, eventName, detail
    } ,const event = new CustomEvent(eventName
      ,detail
      ,bubbles: true
      cancelable: true
      ;({

;(element.dispatchEvent(event)
;return this
{
{

    שימוש //

```

```

;()const eventManager = new EventManager

    throttling עם Event //
} <= () , 'eventManager.on(window, 'scroll'
    ;('גלילה')console.log
    ;({ throttle: 100 } ,{

    debouncing עם Event //
} <= (eventManager.on(searchInput, 'input', (e
    ;(e.target.value , 'חיפוש:')console.log
    ;({ debounce: 300 } ,{

    Event delegation //
} (eventManager.delegate(document.body, '.button', 'click', function(e
    ;(this.textContent , 'כפתור נלחץ:')console.log
    ;({

    Custom event //
} , 'eventManager.emit(document, 'userLoggedIn'
    ,userId: 123
    'יוסי' :userName
    ;({

    custom event לאזנה //
} <= (document.addEventListener('userLoggedIn', (e
    ;(e.detail , 'משתמש התחבר:')console.log
    ;({
    ``

:DOM Performance Optimization ###
    javascript``
    DocumentFragment לביצועים טובים יותר
    } (function createMultipleElements(count
;()const fragment = document.createDocumentFragment

    } (++for (let i = 0; i < count; i
;('const div = document.createElement('div
    ;`{i}$אלמנט` = div.textContent
    ;(fragment.appendChild(div
    {

    (יעיל יותר) DOM-ל אחת //
; (document.body.appendChild(fragment
    {

    Virtual scrolling לרשימות ארוכות //
    } class VirtualList
} (constructor(container, items, itemHeight, renderItem
    ;this.container = container
    ;this.items = items
    ;this.itemHeight = itemHeight
    ;this.renderItem = renderItem
    ;this.visibleStart = 0
    ;this.visibleEnd = 0

    ;()this.init
    {

```

```

    } ()init
this.container.style.height = `${this.items.length *
    ;`this.itemHeight}px
    ;'this.container.style.position = 'relative
    ;'this.container.style.overflow = 'auto

    ;('this.viewport = document.createElement('div
    ;'this.viewport.style.position = 'absolute
    ;'this.viewport.style.top = '0
    ;'this.viewport.style.width = '100%

    ;(this.container.appendChild(this.viewport

    } <= () , 'this.container.addEventListener('scroll
    ;()this.updateVisibleItems
    ;({

    ;()this.updateVisibleItems
    {

    } ()updateVisibleItems
    ;const scrollTop = this.container.scrollTop
    ;const containerHeight = this.container.clientHeight

    ;(this.visibleStart = Math.floor(scrollTop / this.itemHeight
    )this.visibleEnd = Math.min
    this.visibleStart + Math.ceil(containerHeight /
    ,this.itemHeight) + 1
    this.items.length
    ;(

    ;()this.render
    {

    } ()render
    ;' = this.viewport.innerHTML
this.viewport.style.transform = `translateY(${this.visibleStart *
    ;`this.itemHeight}px

    } (++for (let i = this.visibleStart; i < this.visibleEnd; i
    ;(const item = this.renderItem(this.items[i], i
    ;`item.style.height = `${this.itemHeight}px
    ;(this.viewport.appendChild(item
    {
    {
    {

    שיחוש //
:const items = Array.from({ length: 10000 }, (_, i) => ({ id: i, name
    ;(({ `i}$ פריט`

    )const virtualList = new VirtualList
    ,('document.getElementById('list-container
    ,items
    גובה פריט // ,50
    } <= (item, index)
    ;('const div = document.createElement('div
    ;div.textContent = item.name

```

```

        ;'div.style.padding = '10px
; 'div.style.borderBottom = '1px solid #ccc
        ;return div
    {
        ;(
        ;\`
    }

```

מחלקת Import/Export-1 Modules ##

```

        :ES6 Modules ###
        javascript``
        מודול math.js //
;export const PI = 3.14159
;export const E = 2.71828

    } (export function add(a, b
        ;return a + b
        {

    } (export function multiply(a, b
        ;return a * b
        {

    } export class Calculator
        { ()constructor
        ;this.result = 0
        {

            } (add(num
;this.result += num
        ;return this
        {

            } (multiply(num
;this.result *= num
        ;return this
        {

            } ()get value
;return this.result
        {
            {

                Default export //
    } (export default function subtract(a, b
        ;return a - b
        {

            או כולם יחד //
;{ const constants = { PI, E
;{ const functions = { add, multiply

;{ export { constants, functions
        ;\`

        javascript``
        שימוש במודול - main.js //

```

```

Named imports //
; 'import { PI, add, multiply, Calculator } from './math.js

Default import //
; 'import subtract from './math.js

שילוב //
; 'import subtract, { PI, add } from './math.js

הכל Import //
; 'import * as math from './math.js

Rename imports //
; 'import { add as sum, multiply as product } from './math.js

Re-export //
; 'export { PI, add } from './math.js
; 'export { default as subtract } from './math.js

Dynamic imports //
    } () async function loadMath
; ('const mathModule = await import('./math.js
    ; return mathModule
    {

Conditional imports //
    } (if (someCondition
; ('const { heavyFunction } = await import('./heavy-module.js
    ; () heavyFunction
    {

(HTML) Import maps //
    */
    <"script type="importmap>
    }
    } : "imports"
, "lodash": "https://cdn.skypack.dev/lodash"
  "react": "https://cdn.skypack.dev/react"
    {
    {
    <script/>
    /*
    ...

:Module Patterns ###
  javascript``
  Singleton Pattern //
} class DatabaseConnection
    } () constructor
    } (if (DatabaseConnection.instance
; return DatabaseConnection.instance
    {

    ; this.connection = null
; DatabaseConnection.instance = this
    {

    } () connect

```

```

        } (if (!this.connection
;{ 'this.connection = { status: 'connected
        ;('מתחבר למסד נתונים')console.log
        {
            ;return this.connection
        }
    }

;()export default new DatabaseConnection

        Factory Pattern //
    } class UserFactory
    } (static createUser(type, data
        } (switch (type
            : 'case 'admin
; (return new AdminUser(data
            : 'case 'regular
; (return new RegularUser(data
            : 'case 'guest
; (return new GuestUser(data
            : default
; ({throw new Error(`Unknown user type: ${type
        {
            {
                {

;{ export { UserFactory

        Observer Pattern //
    } class EventEmitter
    } ()constructor
;{} = this.events
    {

        } (on(event, callback
    } ([if (!this.events[event
; [] = [this.events[event
    {
; (this.events[event].push(callback
    {

        } (off(event, callback
    } ([if (this.events[event
this.events[event] = this.events[event].filter(cb => cb !==
; (callback
    {
        {

        } (emit(event, data
    } ([if (this.events[event
; ((this.events[event].forEach(callback => callback(data
    {
        {
            {

;{ export { EventEmitter
    ``

```

```

        ## Web APIs מתקדמים

        ### Fetch API מתקדם
        javascript``
        } class HttpClient
    } ({}) = constructor(baseUrl = '', defaultHeaders
        ;this.baseUrl = baseUrl
        } = this.defaultHeaders
        , 'Content-Type': 'application/json'
        defaultHeaders...
        ;{
        } = this.interceptors
        , [] : request
        [] : response
        ;{
        {

        Request interceptor //
        } (addRequestInterceptor(interceptor
        ;(this.interceptors.request.push(interceptor
        {

        Response interceptor //
        } (addResponseInterceptor(interceptor
        ;(this.interceptors.response.push(interceptor
        {

        } ({}) = async request(url, options
        Apply request interceptors //
        } = let config
    , { headers: { ...this.defaultHeaders, ...options.headers
        options...
        ;{

    } (for (const interceptor of this.interceptors.request
        ;(config = await interceptor(config
        {

        const fullUrl = url.startsWith('http') ? url :
        ;`${this.baseUrl}${url}

        } try
        ;(let response = await fetch(fullUrl, config

        Apply response interceptors //
    } (for (const interceptor of this.interceptors.response
        ;(response = await interceptor(response
        {

        } (if (!response.ok
        throw new Error(`HTTP ${response.status}:
        ;`${response.statusText}
        {

        ;('const contentType = response.headers.get('content-type
        (('if (contentType && contentType.includes('application/json
        }

        ;()return await response.json
    }

```

```

        {
            ;()return await response.text
                } (catch (error {
; (console.error('Request failed:', error
                    ;throw error
                        {
                            {
                                } ({) = get(url, options
;({ return this.request(url, { method: 'GET', ...options
                                    {
                                        } ({) = post(url, data, options
                                        } ,return this.request(url
                                            , 'method: 'POST
                                            , (body: JSON.stringify(data
                                                options...
                                                    ;({
                                                        {
                                                            } ({) = put(url, data, options
                                                            } ,return this.request(url
                                                                , 'method: 'PUT
                                                                , (body: JSON.stringify(data
                                                                    options...
                                                                        ;({
                                                                            {
                                                                                } ({) = delete(url, options
;({ return this.request(url, { method: 'DELETE', ...options
                                                                                    {
                                                                                        {
                                                                                            שיחון //
; ('const api = new HttpClient('https://api.example.com
                                                                                      הוספת //
                                                                                      interceptors
} <= (api.addRequestInterceptor(async (config
; ('const token = localStorage.getItem('authToken
    } (if (token
; `{config.headers.Authorization = `Bearer ${token
    {
        ;return config
    ;({
        } <= (api.addResponseInterceptor(async (response
            } (if (response.status === 401
                Redirect to login //
; 'window.location.href = '/login
    {
        ;return response
    ;({
        ``
        :Session Storage-1 Local Storage ###
        javascript``
        } class StorageManager

```

```

    } (constructor(storage = localStorage
        ;this.storage = storage
    {

        } (set(key, value, expiry = null
            } = const item
            ,value
        expiry: expiry ? Date.now() + expiry : null
        ;{
        ;((this.storage.setItem(key, JSON.stringify(item
        {

            } (get(key
        ;(const itemStr = this.storage.getItem(key
        ;if (!itemStr) return null

        } try
        ;(const item = JSON.parse(itemStr

        בדיקת תפוגה //
    } (if (item.expiry && Date.now() > item.expiry
        ;(this.storage.removeItem(key
        ;return null
        {

        ;return item.value
        } (catch (error {
; (console.error('Error parsing stored item:', error
        ;return null
        {

        } (remove(key
        ;(this.storage.removeItem(key
        {

        } ()clear
        ;()this.storage.clear
        {

        } ()keys
        ;(return Object.keys(this.storage
        {

        Cache with automatic cleanup //
        setCache(key, value, ttl = 3600000) { // 1 hour default
        ;(this.set(`cache_${key}`, value, ttl
        {

        } (getCache(key
        ;(`{return this.get(`cache_${key
        {

        } ()clearExpired
        ;()const keys = this.keys
        } <= keys.forEach(key
        זה יסיר פריטים שפגו // ;(this.get(key
        ;({

```

```

    {
    {

        שימוש //
        ;()const storage = new StorageManager
;(const sessionStorage = new StorageManager(window.sessionStorage

        שמירה עם תפוגה (שעה) //
;(age: 25 }, 3600000 , 'יוסי' :storage.set('userData', { name

        קבלה //
;('const userData = storage.get('userData

        Cache //
storage.setCache('apiResponse', data, 600000); // 10
;('const cachedData = storage.getCache('apiResponse
``,`

```

זהו חלק נרחב נוסף מהמדריך. האם תרצה שאמשיך עם עוד נושאים כמו Web Workers, Service Workers, WebSockets או נושאים אחרים?