

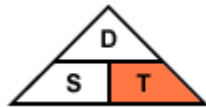
Coding Basics

Setup

- Use the same exercise-runner project for all the exercises below
- For each exercise, create its own file, named as: 01.js, 02.js...
- Copy the question as a comment at the top of your javascript file

Basics

1. Read first name and last name, also create a variable `fullName`, and then welcome the user by his full name.
2. Read 2 numbers and print the result of the following operations on them: `(*, /, %)`
3. Read temperature in Celsius, and print in Fahrenheit
4. Read distance and speed and print ETA



$$\text{Time} = \frac{\text{Distance}}{\text{Speed}}$$

(Now let's build waze ;))

5. Rolling Project - BankSystem (you will add more features to this project as we go along)
 - Initialize a variable: `currentBalance` with the value: 1000
 - prompt for the user secret and how much he would like to withdraw, and then print a nice message with the new balance.
6. Ask the user for 3 digits and print the number in full
for example: if the user entered 3,2,6 then we print 326.
 - BONUS: In this case, working with strings is easier, try solving using numeric variables
7. BONUS: read `a`, `b`, `c` from the user, (for example: 2, -5, 2)

your output should be something like:

$$2x^2 - 5x + 2 = 0$$

$$x1 = 2 ; x2 = 0.5$$

For being able to print the 2 exponent to the console (X Baribua...)
google for:

unicode exponent two and copy the character to your IDE

Conditions

8. Read 3 numbers and check if the 3rd is the sum of the first two, if so, print all numbers to the console like this: $6 + 4 = 10$
9. Read 3 numbers and print the smallest
10. Read 2 numbers and calculate the difference (absolute value)
 - if the diff is smaller from both values say that those numbers are relatively-close
 - Validate that you got numbers
(hint: search for something like: *javascript check if number*)
11. Ask the user how many friends he has on FB and print out an analysis:
 - More than 500 – OMG, celebrity!
 - More than 300 (and up to 500) – Well connected!
 - 100 and more – Knows some people
 - Up to 100 – Quite picky aren't you?
 - 0 – Lets be friends?
12. Rolling Project: BankSystem
 - Check that the secret is 'secret!', if not, tell the user its wrong and don't let him withdraw
 - Add feature: don't let the user withdraw more than he has in the account
13. Guess Who
 - Alert the user to think about some actor
 - Using the *confirm* function, ask the user 2 yes-no questions:
 - Male?
 - Blond?
 - Yes: Philip Seymour!
 - No: Tom Cruise!
 - English?
 - Yes: Julianne Moore!
 - No: Natalie Portman!
14. The Elevator –
 - Keep a currentFloor variable, initialize it to 0
 - Ask the user which floor he needs
 - Validate its between -2 (parking) and 4

- Update the currentFloor variable accordingly
- If the user goes to 0 say 'Bye Bye'
- If the user goes to parking output: 'Drive Safely'

Strings

15. Read 2 names and print the longer one

16. read a string and print:

- Its length
- Its first and last chars
- in uppercase and lowercase

17. Rolling Project: BankSystem

- Make the secret case insensitive

Functions

18. Write a function that gets a name as a parameter and greets him

19. Write a function that gets 2 numbers and return their sum

20. Write a function *isEven* that gets a number, and returns *true* if the number is even otherwise *false*.

21. Write a function *getBigger* that receives 2 numbers and returns the bigger.

22. Write a function: *isOfAge* that gets a name and an age, if the user is not yet 18, alert him that he is too young, this function also returns a boolean

Loops

23. Read 10 numbers from the users and output "the number is even" if this is the case, else "the number is not even"

24. Read 10 numbers and print:

- The maximum number
- The minimum number
- The average

25. Read numbers until 999, for each number:

- Print if it's divided by 3
- Print whether this num is much bigger (more than 10) than the previous number.

26. Write the function *myPow* that gets 2 params: *base*, *exp* and returns the power.
27. Write a function *getAsterisks* which gets a number, and returns a string containing asterisks according the number supplied.
- Write a program that prints in the first line 1 asterisk, in the second 2 asterisk and so on until 10
 - After reaching a line with 10 asterisks, and goes backwards until a line with 1 asterisk
 - There is a new requirement to support any character (not necessarily asterisk), refactor your code, the character will be read from the user and be sent to the function.
 - Use `Math.rand` to generate a number between 1 and 10 and use it to draw asterisks rows at random lengths
28. Write the function *factorial* that gets a number *n* and return *n!*
29. Write a function that prints the multiplication table
30. Let the user guess a random number your program set, help the user but outputting Too Low or Too High.
31. Write a program that generates 30 random numbers.
The numbers should be generated so each number is greater than the ones generated before.
32. Write a program to compute the greatest common divisor (GCD) of two positive integers.
33. Write a function that prints a block of asterisks (*) by parameters: rows, cols
ex: for 4,5.

```
*****
*****
*****
*****
```

Now, print only the outline

```
*****
*      *
```

```
*      *  
*****
```

34. Rolling Project – BankSystem:

- Improve the secret code coming from user, instead of hardcoded secret - symmetric codes (such as: 95459) are accepted.

35. Read a number and

- Calculate sum of its digits.
- Calculate multiply of its digits.
- Print it with swapped first and last digits (2731 => 1732)
- Print its reverse (BONUS: as number not as string).
- Sum its first and last digit.
- Print if it's an Armstrong number.
i.e. 371 is an Armstrong number:
 $3^3 + 7^3 + 1^3 = 371$
- Print if it's a perfect number.
Perfect number is number that sum of all his dividers is the number itself.
i.e. 6 is a perfect number (1 + 2 + 3).

Now read a number: *max* and prints all Armstrong and Perfect positive numbers until *max*

36. Write the function myAbs() to return absolute values, the compare the performance of your function and the original function (by running them 1000000 times each).

37. Write A function that generates a random number between 0 and *max* (Parameter)

Clue: Use Math.Random & Math.Floor

Strings and Loops

38. Read a string and print it backwards using loop

39. VOWELS (aoieu)

- write a function that gets a string and print how many times each vowel appears
 - Read a string and make the vowels lowercase, the others uppercase (GiZiM GiDoo)
 - Write a function that reads a string and double all the vowel in it
40. write a function that accepts 2 strings, and returns the location of the second string in the first, if the string is not found return false
41. Write the function encrypt that gets a string and encrypt it by replacing each character c with c+5 (i.e. 'r' will be replaced by: 'w'), now write a function to decrypt a message.
42. Write 2 Functions that get a comma ',' delimited string that consists of names. Ex: 'igal,moshe,haim' and returns who has the longest name, and the shortest name. Tip: use the *split* function to covert a string to an array
43. Write a function: generatePass(passLength) that generates a password of a specified length. Password is made out of random single-digit numbers, using our first function.

Arrays

44. Write a function countApperance that gets an array: names and a variable name and counts how many times name appears in names
45. Write a function biggerThan100 that gets an array of nums and returns an array of items that are bigger than 100.
46. Create an array of all characters in english. Write a function that gets a length and return a sentence with made-up words (lorem ipsum...)
47. Read a number and print each number in words.
I.E – 123 – One Two Three. 7294 – Seven two nine four
48. Write a function startWithS that gets an array of names and returns an array of names that start with S
- a. Improve that to work on any letter by adding a letter argument

49. Write a function that gets 2 arrays and returns a sum array:
- b. For example, for: [1, 4, 3] [2, 5, 1, 9] it returns: [3, 9, 4, 9]
 - c. Read those arrays from the user (until 999 is entered)
50. Write a function: findUniques that gets an array: nums with numbers in the range 0 -10, and return a modified array where each value appear only once.
- d. Find the unique elements from two arrays and insert them into a new array that should also hold unique only.
51. Write a function: multBy that gets an array: nums and a num and returns a modified array in which each item in the array is multiplied by num.
- e. Add another param: immutable, when true use slice to work on a new array and leave the original array as is.
52. Write a program that reads nums in the range 0 -3. The loops should stop when 999 is entered. Then print how many times each of these nums was entered.
53. Implement your own version of the split function. Test with different types of data, measure time difference between your function and String.split
54. Write a function to get nth largest element from an unsorted array, without sorting the array.
- For example: nthlargest([43, 56, 23, 89, 88, 90, 99, 652], 4)
- Result: 89
55. Implement bubble sort for array, measure time difference between your function and Array.sort
56. Write a function that get some text and return an associative array in which the key is a word and the value is the count (how many times this word appeared in the text)
57. Write a function that gets a hebrew letter and returns its GIMATRIA value.
58. Implement lion in desert for a sorted array (WTF)

59. Pick random atoms from an array, how much time passes before you get water (H₂O)?

Multi-Dimensional Arrays

60. Generate the multiplication table in a 2d array

61. Write a function that returns the maximum value in a 2d array.

62. Fill up a 2d array with random numbers in a specified range.

Create functions that create excel-like functions on it: count, sum, max, min, average

63. Write a function named findMode that will find and print, the number that occurs most often in a 2d array.

BONUS: If there are ties, print all

64. Write a function that gets a 2d array and checks if it is a magic square:

- it must be square,
- Rows' sums, columns' sums, and the two diagonals sums should be equal.

Example:

2	7	6	→15
9	5	1	→15
4	3	8	→15
↙15	↓15	↓15	↓15
			↘15

65. BONUS: Create a 2 dimensional array (X*Y) filled randomly with Zeros and Ones.

Example:

```
[1][0][1][0][1][0]
[0][1][1][0][0][0]
[0][0][1][0][1][1]
[0][0][0][0][1][1]
[0][0][1][1][0][0]
```



```
[1][0][0][1][1][1]
```

Write a function called `shiftLeft` which moves all the ones to the left like so:

```
[1][1][1][0][0][0]
[1][1][0][0][0][0]
[1][1][1][0][0][0]
[1][1][0][0][0][0]
[1][1][0][0][0][0]
[1][1][1][1][0][0]
```

Expand the function to get a parameter which decides which way the ones move (up,down,left,right)

Objects

66. Monsters

- Create a `monsters` array
- Read name from the user
- Each monster should also have power (random 1-10)
- Write the function: `findMostPowerful(monsters)`

67. Students

- Create a `students` array
- Read student's name until quit and populate the `students` array with student objects
- Read 3 grades for each student
- Calc avg for each student
- Write the function `findWorstStudent(students)`
- Write the function `sortStudentsByGrade(students)`
- Write the function `sortStudentsByName(students)`

68. Airline

- a. Build a data structure for an airline:
- b. plane
 - i. sitsCount
- c. passenger
 - i. id (7 digits random number)
 - ii. fullName
 - iii. flights (array)
- d. flight
 - i. at (date)
 - ii. src (origin airport)
 - iii. dest
 - iv. plane (pointer to a plane)
 - v. passengers (array)
- e. Initialize with consistent data (data that makes sense):
- f. Create an array with 5 passengers
- g. Create an array of 2 planes
- h. Create an array with 2 flights, each flight has a plane property that points to a plane object, and an array of pointers to passengers
- i. Write functions:
 - i. isFlightFullyBooked(flight)
 - ii. getFrequentFlyer ()

69. Metushelah

BONUS: Build a data structure that represent a family tree, we might make it graphical later, but for now we focus on the data structure:

- a. A person object that has:
 - i. an id (7 chars random string),
 - ii. a name,
 - iii. a birthdate (a date object)
 - iv. parents – array by size of 2
 - v. childs – array by any size
- b. Build a data structure byID that will store all people by their ID
- c. write functions:
 - i. addChild(toPersonId, childPerson)

ii. `addParent(toPersonId, parentPerson)`

- d. Use those functions to create a tree with some data (i.e. your family, from Adam to Noah, etc.)
- e. Write a function that prints the name of the parents that gave their son, the longest name.

70.Symmetric Matrix

מטריצה סימטרית היא מטריצה שבה מתקיים: $A[i,j]=A[j,i]$.
 א. תן/י דוגמה למטריצה סימטרית בגודל 3×3 .
 ב. כתב/י שגרה המחזירה אמת כאשר המטריצה היא סימטרית.

71.Bingo

בערב בינגו חילקו טפסים המכילים 25 מספרים אקראיים בין 1 ל-99.
 הכנס/י ערכים אקראיים לשני טפסים שחולקו במסיבה והדפס את הטפסים.
 כתב/י תוכנית שמדמה את רשימת המספרים שהוגרלו במסיבה עד לקבלת טופס מנצח ראשון.
 הדפס/י מי ניצח.
 הצג בממשק HTML

72.Latin Square

"ריבוע לטיני" הוא ריבוע המכיל מספרים כך שכל מספר לא מופיע יותר מפעם אחת בשורה או בעמודה שלו. כתוב פונקציה המקבלת מטריצה בגודל $N \times N$ ומחזירה אמת כאשר המטריצה היא ריבוע לטיני.

73.Magic Square

ריבוע קסם הוא לוח המכיל מספרים שלמים (1 עד N^2) בגודל $N \times N$ (N אי זוגי) בו סכום המספרים בכל שורה, עמודה ואלכסון שווה.

כתוב שגרה המקבלת מערך דו ממדי בגודל $N \times N$ ומחזירה אמת כאשר המערך מהווה ריבוע קסם.

בכדי ליצור ריבוע קסם יש לבצע את התהליך הבא :
 א. מקם את המספר 1 באמצע השורה העליונה.
 ב. עלה משבצת אחת למעלה, ומשבצת אחת שמאלה, ומקם את המספר הבא. כל מספר נוסף ימוקם ע"פ תהליך זה.
 ג. אם התהליך הנ"ל גורם לחריגה מהלוח, יש לעבור לצידו ההפכי של הלוח : חריגה מלמעלה תגרום להופעה למטה, חריגה משמאל תגרום להופעה בצידו הימני של הלוח.
 ד. אם המיקום החדש תפוס, יש לרדת משבצת אחת מהמקום הקודם.
 כתוב תוכנית בה הקלט הוא מספר המציין את גודל הריבוע (בין 3 ל-15), והפלט הוא ריבוע קסם הנבנה ע"פ התהליך שתואר.

74.Supermarket

בחנות סופרמרקט ישנן 7 קופות ו-10 מחלקות. לגבי כל קנייה רושמים את המספר הסידורי של הקופה, את מחיר הפריט ואת המספר הסידורי של המחלקה שאליה שייך הפריט. בסוף כל יום אוספים את הרישומים האלה מכל 7 הקופות.
כתב/י תוכנית שתקבל כקלט את אוסף הרישומים האלה ותקבע:
א. את מספר הפריטים שנמכרו מכל מחלקה ומחלקה ואת שווים הכספי הכולל.
ב. את הפדיון הכספי הכולל של כל קופה וקופה.
מאיזו מחלקה נמכרו סחורות בערך הכספי הגדול ביותר.

75.SimCity

טבלת המרחקים הבאה מכילה את המרחק בין כל שתי ערים:

	A	B	C	D
מספר עיר				
A	0	18	10	12
B	18	0	17	20
C	10	17	0	13
D	12	20	13	0

(למשל המרחק בין עיר A לעיר C הוא 10 ק"מ).

- כתוב פונקציה המקבלת טבלת מרחקים ועיר ומחזירה שם העיר הרחוקה ממנה ביותר ואת המרחק בניהן.
- כתוב פונקציה המדפיסה עבור כל עיר את שם העיר הכי רחוקה ממנה.
- מעוניינים להקים תחנת מגן דוד אדום המשותפת לכל 4 הערים. התחנה תמוקם בעיר שמרחקה המקסימלי מכל עיר אחרת הוא הקטן ביותר. כתוב פונקציה בה הקלט הוא מטריצת המרחקים ומדפיסה את העיר בה תמוקם התחנה.

76.Sparse Matrix

מטריצה דלילה היא מטריצה ריבועית ($N \times N$) בה מתקיימים שני התנאים הבאים :
 I) מספר איברי המטריצה השונים מאפס הוא פחות מעשרים אחוז מאיברי המטריצה.
 II) בכל שורה יש לכל היותר שני איברים השונים מאפס.
 למשל, המטריצה הבאה היא מטריצה דלילה :

0	0	0	9	0
0	0	0	0	0
0	0	8	0	0
0	0	25	0	0
0	0	0	0	15

א. כתבי/י שגרה המקבלת מטריצה ומחזירה אמת כאשר המטריצה היא מטריצה דלילה.
 ב. מטריצה דלילה ניתן לשמור בזיכרון באמצעות שלושה מערכים חד ממדיים, כאשר המערך הראשון מכיל את איברי המטריצה השונים מאפס. שני המערכים האחרים מכילים את האינדקסים של השורה והעמודה של איברים אלה.
 לדוגמא, שלושת המערכים שישמרו את המטריצה הדלילה הנ"ל הם :

מערך ערכים

15	25	8	9
----	----	---	---

מערך שורות

4	3	2	0
---	---	---	---

מערך עמודות

4	2	2	3
---	---	---	---

כתבי/י שגרה המקבלת כפרמטרים את שלושת המערכים ואת מספר איברי המטריצה המקורית, ומחזירה את המטריצה הדלילה בצורתה המקורית.

77.Game of Life

המשחק אמור לתאר לידה ומוות של יצורים, כאשר הלידה והמוות נקבעים לפי כללים מסוימים.
 שדה המשחק : לוח משבצות שגודלו נקבע על פי השחקן (המשתמש).
 יצור חי יתואר ע"י X. אם אין יצור חי המשבצת תישאר ריקה.
 כל התאים הסמוכים למשבצת מסוימת הם השכנים שלה (כלומר לכל היותר 8 שכנים).

			X	X			X
X		X		X	X	X	
			X	X		X	

	X						
					X		
		X		X	X		
X	X		X		X	X	

כללי המשחק :

1. במשבצת שלה 0-2 שכנים שהם יצורים חיים, לא יתכנו חיים בדור הבא (כלומר : אם היו בה חיים היצור ימות מבדידות).
2. במשבצת שלה 3-5 שכנים שהם יצורים חיים, יתכנו חיים בדור הבא (כלומר : אם לא היה בה יצור – ייוולד חדש, ואם היה – הוא ישאר בחיים).
3. במשבצת שלה 6-8 שכנים שהם יצורים חיים, לא יתכנו חיים בדור הבא (כלומר : אם היו בה חיים היצור ימות מצפיפות).

הקלט : גודל שדה המשחק ומספר הדורות (מספר מהלכי המשחק).
הפלט : מצב היקום בכל דור ודור.