50+ Javascript Practice Exercises

Beginners to Advanced

*Beginner Level:

Console Output:

Print "Hello, World!" to the console.

*Beginner Level:

Looping:

Print the numbers from 1 to 5 to the console using a loop.

*Beginner Level:

Functions:

Write a function to add two numbers and return the result.

*Beginner Level:

Functions:

Create a function to calculate the area of a rectangle given its width and height.

*Beginner Level:

String Manipulation::

Write a function that takes a string and returns the reversed version of it.

*Beginner Level:

Conditional::

Write a function that checks if a number is even or odd and returns "Even" or "Odd" accordingly.

*Beginner Level:

Conditional::

Create a program that checks if a given year is a leap year.

*Beginner Level:

Arrays and Loops:

Find the sum of all elements in an array.

*Beginner Level:

Objects*

Create an object representing a car with properties like make, model, and year.Add a method to the car object to start the engine.

*Beginner Level:

Basic DOM Manipulation:

Change the text of a paragraph on a webpage using JavaScript.

*Beginner Level:

Basic DOM Manipulation:

Create a button dynamically and add it to the webpage.

*Beginner Level:

Event Listeners:

Add a click event listener to a button that displays an alert when clicked.

*Beginner Level:

Functions and Scope:

Write a function that calculates the factorial of a given number.

*Beginner Level:

Functions and Scope:

Create a function that generates a random number between a given range.

*Beginner Level:

Higher-Order Functions:

Write a function that takes an array of numbers and returns a new array with only the even numbers.

*Beginner Level:

Higher-Order Functions:

Use `map()` to double all the elements in an array.

*Beginner Level:

Array Methods:

Find the largest element in an array using the `reduce()` method.

*Beginner Level:

Array Methods:

Remove all occurrences of a specific element from an array.

*Beginner Level:

*Date and Time

Create a function that displays the current date and time in a specific format.

Intermediate Level:

Error Handling:*

Implement a try-catch block to handle an error that occurs during API data fetching.

Intermediate Level:

Recursion

Write a recursive function to calculate the factorial of a given number.

Intermediate Level:

Recursion

Implement a recursive function to find the nth Fibonacci number.

Intermediate Level:

*Closures

Create a counter function using closures that increments and returns the count on each call.

Intermediate Level:

*Closures

Implement a private variable using closures.

Intermediate Level:

Prototypes:*

Create a prototype for a Product object with properties like name, price, and quantity. Add a method to the Product prototype to calculate the total value.

Intermediate Level:

Async Programming - Callbacks:*

Implement a function that makes multiple API calls and processes the data using callbacks.

Intermediate Level:

Async Programming - Promises:*

Rewrite the previous exercise using Promises.

Intermediate Level:

Async Programming - Promises:*

Use Promises to load multiple images asynchronously and display them on a webpage.

Intermediate Level:

Async Programming - Async/Await:*

Rewrite the previous exercise using async/await.

Intermediate Level:

Async Programming - Async/Await:*

Implement an async function to fetch data from an API and handle errors using try/catch.

Intermediate Level:

DOM Manipulation - Creating Elements:*

Create an image gallery using dynamically generated elements.

Intermediate Level:

DOM Manipulation - Form Validation:*

Implement a form validation function that checks if all required fields are filled out.

Intermediate Level:

Event Bubbling and Capturing:*

Create multiple nested elements and observe the event bubbling and capturing behavior.

Intermediate Level:

Event Bubbling and Capturing:*

Implement a click event on a parent element that triggers different actions based on which child element was clicked.

Intermediate Level:

Set Interval:*

You are tasked with creating a countdown timer using JavaScript and the `setInterval` function. The countdown timer should start at 1 minute (60 seconds) and update every second until it reaches 00:00. When the countdown reaches zero, an alert should pop up to notify that the countdown is completed.

Intermediate Level:

Set Timeout:*

Write the JavaScript code to implement the quote-changing feature

Advanced Level:*

Regular Expressions:*

Write a regular expression to validate an email address.

Advanced Level:*

Multidimensional Array:

Create a JavaScript function that generates a 2D array with the specified number of rows and columns. Each element in the array should contain the sum of its row index and column index. Once you've created the array, write two additional functions to display the array in its original form and in reverse.

Advanced Level:*

Destructuring:*

Destructure an object to get its properties.

Advanced Level:*

Class:*

Create a class representing a Book with properties like title, author, and year.Add a method to the Book class to get the book's age (current year - year of publication).

Advanced Level:*

Inheritance and Prototypes:*

Create a subclass `Magazine` that extends the `Book` class with an additional property `issue`.Add a method to the `Magazine` class to get the magazine's issue number.

Advanced Level:*

Prototypal Inheritance:*

Create an object `person` with properties like name and age. Then, create a new object `student` that inherits from `person` and has an additional property `studentId`.Add a method to the `person` object and demonstrate that `student` also has access to it.

Advanced Level:*

Random Exercise*

Create a function that determines whether a number is **Oddish** or **Evenish**. A number is **Oddish** if the sum of all of its digits is odd, and a number is **Evenish** if the sum of all of its digits is even. If a number is **Oddish**, return "Oddish". Otherwise, return "Evenish".

Advanced Level:*

Random Exercise*

Given a string, reverse all the words which have odd length. The even length words are not changed.

Advanced Level:*

Random Exercise*

Create a function that takes an array of objects (groceries) which calculates the total price and returns it as a number.

```
// 1 bottle of milk & 1 box of cereals:
getTotalPrice([
  { product: "Milk", quantity: 1, price: 1.50 },
  { product: "Cereals", quantity: 1, price: 2.50 }
 ]) → 4
```

Advanced Level:*

Random Exercise*

Create a function that takes an array of numbers and return "Boom!" if the digit 7 appears in the array. Otherwise, return "there is no 7 in the array".

```
sevenBoom([1, 2, 3, 4, 5, 6, 7]) → "Boom!"

  // 7 contains the number seven.
sevenBoom([2, 55, 60, 97, 86]) → "Boom!"

  // 97 contains the number seven.
```

Advanced Level:*

Random Exercise*

Create a function that takes a string as an argument. The function must return a string containing 1s and 0s based on the string argument's words. If any word in the argument is not equal to "zero" or "one" (case insensitive), you should ignore it. The returned string's length should be a multiple of 8, if the string is not a multiple of 8 you should remove the numbers in excess.

```
textToNumberBinary("Zero one zero ONE zero one zero one")  → "01010101"

textToNumberBinary("zero one zero one zero one zero one one two")  → ""

textToNumberBinary("zero one zero one zero one zero three")  → ""
```

Advanced Level:*

Random Exercise*

Create a function that takes in a sentence and returns a running list of **consonants per word** and **vowels per word**.

```
stringCode("Happy Birthday To Me!")

➞ ["4 6 1 1", "1 2 1 1"]


// Consonants - 4 : [H, p, p, y], 6 : [B, r, t, h, d, y], 1: [T], 1 : [M]

  // Vowels - 1: [a], 2 : [i, a], 1: [o], 1: [e]
```

Advanced Level:*

Random Exercise*

Create a function that takes the month and year (as integers) and returns the number of days in that month.

```
days(2, 2018) → 28

days(4, 654) → 30
```

Advanced Level:*

Random Exercise*

Given two integers representing the numerator and denominator of a fraction, return *the fraction in string format.*

Input: numerator = 1, denominator = 2

Output: "0.5"