JavaScript Exercise

Ternary Operators:

Q1 Write a function that uses a ternary operator to determine if a given number is even or odd?

//1 Write a function that uses a ternary operator to determine if a given number is even or odd?

function isEvenOdd(num) {

  const checkEvenOdd = num % 2 == 0 ? "even" : "odd";

  return checkEvenOdd;

}

console.log(isEvenOdd(23));

console.log(isEvenOdd(24));

console.log(isEvenOdd(0));

console.log(isEvenOdd(1));

Q2 Write a function to determine if a character string entered is longer than 10 characters.

function countChar(str) {

  var strr = "";

  count = 0;

  for (var i = 0; i < str.length; i++) {

    count += 1;

  }

  return count;

}

function checkChar(str) {

  var countstr = countChar(str);

  if (countstr > 10) {

    return "The character string is greater than 10";

  } else {

    return "The character string is less than 10.";

  }

}

console.log(checkChar("SHDHSJS112323"));

console.log(checkChar("123458"));

Q3 Take the marks of a student as an input. Return first division if they are greater than 60 else return second division.

function checkMarks(num) {

  if (num > 60) {

    return "First Division!";

  } else {

    return "Second Division";

  }

}

console.log(checkMarks(70));

console.log(checkMarks(40));

Q4 Take a number and return the string ‘FizzBuzz’ if it is divisible by both 3 and 5, else return Fizz if it is divisible by 3 and return Buzz if it is divisible by 5 only. Else return the number.(Hint: chaining)

function checkNum(num) {

  if (num % 3 === 0 && num % 5 === 0) {

    return "FizzBuzz!";

  } else if (num % 3 === 0) {

    return "Fizz";

  } else if (num % 5 === 0) {

    return "Buzz";

  } else {

    return num;

  }

}

console.log(checkNum(2));

console.log(checkNum(15));

console.log(checkNum(12));

console.log(checkNum(25));

console.log(checkNum(1));

Scope and Closure:

Q1 Global and Local Scope: Create a variable outside a function and another variable using var with the same name inside a function. Try to access both variables from inside and outside the function. This will help you understand the concept of global and local scope.

Now try the above using let and see what happens.

//Q1 Global and Local Scope: Create a variable outside a function and another variable using var with the same name inside a function. Try to access both variables from inside and outside the function. This will help you understand the concept of global and local scope.

//Now try the above using let and see what happens.

let name = "Ali";

function checkName() {

  let name = "Hamza";

  return name;

}

console.log(name);

console.log(checkName());

Q2 Nested functions: At the start of a function, declare a variable. Now return a function inside the current, that uses the above mentioned variable.

//Q2 Nested functions: At the start of a function, declare a variable. Now return a function inside the current, that uses the above mentioned variable.

//Nested functions

function fun1(a) {

  function fun2(b) {

    return a \* b;

  }

  return fun2(2);

}

var returnedVal = fun1(5);

console.log(returnedVal);

var returnedVal = fun1(100);

console.log(returnedVal);

Q3 Create a function with a private variable ‘count’. Make an instance/object of this function. Make functions inside the object to increase and decrease the value of the private variable. Also make a function to return the value of that variable so you can observe the impact on that variable.

//Q3 Create a function with a private variable ‘count’. Make an instance/object of this function. Make functions inside the object to increase and decrease the value of the private variable. Also make a function to return the value of that variable so you can observe the impact on that variable.

function counter() {

  let count = 0;

  function increment() {

    count++;

  }

  function decrement() {

    count--;

  }

  function getCount() {

    return count;

  }

  return {

    increment,

    decrement,

    getCount,

  };

}

const myCounter = counter();

myCounter.increment();

myCounter.increment();

myCounter.increment();

myCounter.increment();

console.log(myCounter.getCount());

myCounter.decrement();

console.log(myCounter.getCount());

Try/catch block:

Q1 Try to divide a number by zero, and return an error message if it does not happen.

//Try to divide a number by zero, and return an error message if it does not happen

function tryCatch(num) {

  try {

    var res = num / 0;

    return res;

  } catch (e) {

    console.log(e);

  }

}

console.log(tryCatch(22));

Q2 Take a json object as an input and try to parse it as a string. Return an error message if unable to do so.

//Q2 Take a json object as an input and try to parse it as a string. Return an error message if unable to do so.

function ParseJson(str) {

  try {

    const obj = JSON.parse(str);

    return obj;

  } catch (e) {

    return "The string is not valid JSON string";

  }

}

var str1 = '{"name": "John", "age": 30, "city": "New York"}';

var str2 = "name";

console.log(ParseJson(str1));

console.log(ParseJson(str2));

Q3 Take an object and try to access its property. If the property does not exist, print out an error message, else return the value of that property.

//Q3 Take an object and try to access its property. If the property does not exist, print out an error message, else return the value of that property.

//creating an object Person

var person = {

  name: "Fatima",

  age: 22,

  hobbies: ["Reading", "Cooking", "Photography"],

};

//accessing objects

try {

  var name = person.name;

  var school = person.school;

  console.log(name);

  console.log(school);

} catch (e) {

  console.log(e);

}

Array Manipulation:

Q1 Find and return the largest and smallest number of an array.

//Q1 Find and return the largest and smallest number of an array.

//find the largest in array

function isMax(arr) {

  var max = arr[0];

  for (var i = 0; i < arr.length; i++) {

    if (arr[i] > max) {

      max = arr[i];

    }

  }

  return max;

}

function isMin(arr) {

  var min = arr[0];

  for (var i = 0; i < arr.length; i++) {

    if (arr[i] < min) {

      min = arr[i];

    }

  }

  return min;

}

var maxx = isMax([2, 3, 90, 87, 34, 10]);

console.log(maxx);

var min = isMin([2, 4, 5, 6, 0, 100]);

console.log(min);

Q2 Write a function that takes an array of objects, where each object has a "name" property, and returns a new array containing only the names of the objects.

//Q2 Write a function that takes an array of objects, where each object has a "name" property, and returns a new array containing only the names of the objects.

//Array's manipulation

let car = [

  {

    name: "Honda",

    plateno: 3449,

    engineType: "Hybrid",

  },

  {

    name: "Mehran",

    plateno: 1234,

    engineType: "Internal combustion",

  },

  {

    name: "Tesla",

    plateno: 336279,

    engineType: "electric",

  },

];

function nameCar(car) {

  var names = [];

  for (var i = 0; i < car.length; i++) {

    names.push(car[i].name);

  }

  return names;

}

console.log(nameCar(car));