# 23 APRIL

JS TEST 1

**Answer all questions in Section A and Section B**

# Section A: Conditional Statements

1. What are conditional statements in Javascript? (2)

**Conditional statements allow a program to make decisions based on certain conditions.**

1. Which of the following is the correct syntax of an **if** statement in Javascript? (1)
   1. If ( age > 18) :
   2. **If ( age > 18 ) {**
   3. If age > 18 {
   4. If age > 18 :
2. How does the **if-else** statement work?

**It checks a condition and runs a block of code if it's conditions true, otherwise it executes the else block.** (2)

1. Can you provide a basic example of an **if-else** statement?

**let age = 20;**

**if (age >= 18) {**

**console.log("You can vote.");**

**} else {**

**console.log("You cannot vote.");**

**}** (4)

1. What is the ternary operator in Javascript?

**The ternary operator is a shorthand way of writing an if-else statement** (2)

1. Write the above conditional statement (question 4) as a ternary operator? (4)

**let age = 20;**

**console.log(age >= 18 ? "You can vote." : "You cannot vote.");**

1. What is an **else-if** statement? (1)
   1. To print a message regardless of the conditions.
   2. To end the program.
   3. **To provide an additional condition if the if statement is false.**
   4. To check only the first condition.
2. Can you write an **if** statement without an **else**? Why or why not?

**Yes, an if statement can exist without an else since it only checks the condition. If the condition is false, the program simply moves on. The else statement is optional.** (2)

1. What will the following code output? (1)

let number = 5;

if (number > 10) { console.log("Greater than 10");

}else if (number === 5) { console.log("Equal to 5");

}else {

console.log("Less than 10");

* 1. “Greater than 10”
  2. **“Equal to 5”**
  3. “Less than 10”
  4. “Syntax error”

1. Which of the following can be used as a condition in an **if** statement? (1)
   1. A string
   2. A number
   3. A Boolean
   4. **All of the above**
2. What are logical operators in Javascript?

**are used to combine conditions.They help control the flow of your program or how your program should run.** (2)

1. Which logical operator is used to combine two conditions and returns true if both (1) conditions are true?
   1. **&&**
   2. ||
   3. !

**D.** ==

1. What is the difference between **==** and **===** inside an **if** statement?

== loose equality – checks value only

=== strict equality –checks value and data type (2)

1. What are the three main logical operators in Javascript? (3)

**&& (AND), || (OR), ! (NOT)**

1. How does the **NOT (** ! **)** operator work? (1)
   1. **It makes a true value false and a false value true.**
   2. It checks if two values are equal
   3. It returns true only if both conditions are true
   4. It converts a number into a Boolean value
2. What is the result of the following expression? (1)

let age = 25;

let hasID = true;

let isBanned = false;

if (age >= 18 && !isBanned && hasID) { console.log("You can enter the event.");

} else {

console.log("Access denied.");

}

**"You can enter the event."**

1. How does the **AND (** &&**)** operator work?

**It returns true only if all conditions are true.** (2)

1. What is the output of this expression? (1)

console.log(true && false)

* 1. true
  2. **false**
  3. undefined
  4. null

1. Write a condition to check if a person can drive. The person must be 18 or older and must have a driving license**.**

**let age = 20;**

**let hasLicense = true;**

**if (age >= 18 && hasLicense) {**

**console.log("You can drive.");**

**} else {**

**console.log("You cannot drive.");**

**}**

1. How does the OR (||) operator work in conditional statements? (1)
   1. It returns true if both conditions are true
   2. It returns true only if both conditions are false
   3. **It returns true if at least one condition is true**
   4. It reverses the Boolean value of the condition

## Marks : 40

**Section B JavaScript Loops Test**

1. Explain the syntax of a for loop and describe two use cases where it is preferred. **[3]**

**for (i = 1; i < 4; i++) {**

**// Code block**

**}**

**Iterating over arrays and executing code a specific number of times.**

1. Explain how a while loop works and provide an example demonstrating its use. **[3]**

**let count = 0;**

**while (count < 5) {**

**console.log(count);**

**count++;**

**}**

**console.log(count)**

**Executes the code block while the condition is true.**

1. How does a do-while loop differ from while and for loops? **[2]**

**A do-while loop executes the code at least once, even if the condition is false initially.**

1. List two strategies to avoid infinite loops in while or do-while loops. **[2]**

**Ensure the condition eventually turns false by using a counter that increments correctly or use a break statement.**

**let count = 0;**

**while (true) {**

**console.log(count);**

**count++;**

**if (count >= 10) { // ✅ The loop stops when count reaches 10**

**break;**

**}**

**}**

1. Compare scenarios where a for loop is more suitable than a while loop. **[3]**

**A for loop is more suitable when you know exactly how many times you need to iterate eg.** counting, iterating over arrays.

**while while is better for loops with unknown iterations.** Used where the condition is defined by the user input.

1. Explain how break works in a loop and provide a use case. **[2]**

**break exits the loop early when a condition is met.**

1. Write a do-while loop that repeatedly asks for user input until a valid integer between 1 and 10 is entered. **[5]**

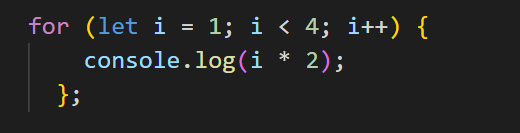
**let num;**

**do {**

**num = prompt("Enter a number between 1 and 10:");**

**} while (num < 1 || num > 10);**

## What is the output of this for loop? [2]

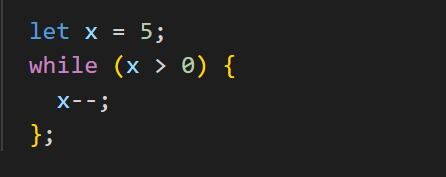


* 1. 1, 2, 3
  2. **2, 4, 6**
  3. 0, 2, 4
  4. 1, 3, 5

## Which loop guarantees at least one execution of its body? [1]

* 1. for
  2. while
  3. **do-while**

## How many times will this while loop run? [1]



* 1. **5**
  2. 4
  3. 6
  4. Infinite

# Javascript Test 2

## 12 May 205

**Complete the questions below, this test has 3 sections (A, B & C) total marks(100)**

## Section A Functions

1. What is a function? (2)

**A function is a reusable block of code that performs a specific task when called.**

1. What is the keyword *return* used for? (1)
   1. **Ends the function and gives back a value.**
   2. Ends the function and logs the result to the console.
   3. Ends the function and pauses the function to resume later.
   4. Ends the function and declares a new variable

**The return statement stops execution of the function and sends back a value to the place where the function was called.**

1. What will the output of this code snippet be? **25** (1)

let square = function(num) { return num \* num;

};

console.log(square(5));

1. Name one of Javascripts built-in Functions?

**Alert(),prompt()** (2)

1. Which symbols surround the body of a function?

**{ }** (1)

1. What part of the function sends something back when a function is called?

**Return statement** (2)

1. At the top of a function declaration, there may be variable names called **parameters** inside the parenthesis(round brackets).When invoking or calling a function, values called **arguments** may be passed- in to be assigned to these variables. (2)
2. True / False: A function can be nested inside another function.

**True**

**function greeting(name){**

**function sayHello(){**

**console.log(“Hello,” + name);**

**} return sayHello;**

**}**

**const greetUser = greeting(“John”);**

**greetUser();**

(1)

1. Expain what is wrong with this code snippet below? (1)

function addNumbers(num1, num2) [ let sum = num1 + num2; console.log(`Sum: ${sum}`);

]

addNumbers(5, 4);

**The function body should be enclosed using {} instead of [].**

1. What is a variable declared outside a function called? (2)

**Global variable**

1. Write a function called ***addThreeNumbers*** that takes **three numbers** as parameters and returns their

**sum**. (6)

function addThreeNumbers(num1, num2, num3) {

return num1 + num2 + num3;

}

console.log(addThreeNumbers(5, 10, 15)); // Output: 30

1. What is the output of this function call? (1)

function greeting(name = *“*Guest*”*) { return *“*Hello, *“* + name;

}

console.log(greeting());

* 1. **Hello, Guest**
  2. Hello,
  3. Undefined
  4. syntax error

1. What are Arrow functions?

**Arrow functions are a shorthand way to write functions in JavaScript using =>**

(2)

1. When a functions calls itself in order to solve a problem its called? (1)
   1. Looping
   2. Returning
   3. **Recursion**
   4. Referencing
2. How do you call a function named ***greet***?

**greet()** (2)

1. What will happen if you call a function without passing in the expected arguments? (1)

function greeting(name ) { console.log(*“*Hello, *“* + name);

}

greeting();

* 1. It will print *Hello*
  2. It will crash the program
  3. **It will print *Hello, undefined***
  4. It will cause a syntax error

1. What is the purpose of parameters in functions?

**Parameters allow functions to take inputs or it olds the value you pass when calling a function eg.No 11**

1. True / False : a ***callback function*** in Javascript is a function that is passed as an argument to another function and is executed after the completion of that function.

**True**, **A callback function is passed into another function and executed after that function finishes.**

**function sayHello(name, callback) {**

**console.log("Hello, " + name);**

**callback(); // Calls the callback function**

**}**

**function goodbye() {**

**console.log("Goodbye!");**

**}**

**// Call the function with a callback**

**sayHello("Chantal", goodbye);**

1. Fill in the blanks to create a function that takes ***two strings*** and returns their concatenation? (3)

function concatenateStrings( , ) { return str1 + str2;

}

Let result = (*“*Java*”* + *“*script*”*) console.log(result);

1. Write a function that will ask the user for their name and then greets them.

**function askName() {**

**let name = prompt("Enter your name:");**

**console.log("Hello, " + name);**

**};** (6)

# SECTION B Arrays

### Multiple Choice Questions (8 marks)

1. What does the push() method do in JavaScript arrays?
   1. Removes the first element
   2. **Adds an element to the end**
   3. Sorts the array
   4. Deletes an element by index
2. Which method is used to remove the last element of an array?
   1. shift()
   2. splice()
   3. **pop()**
   4. unshift()
3. What is the index of the first element in any array?
   1. **0**
   2. 1
   3. -1
   4. Depends on the array
4. Which method adds an element to the beginning of an array?
   1. push()
   2. splice()
   3. **unshift()**
   4. concat()
5. Which of these returns the index of the first occurrence of a value?
   1. **indexOf()**
   2. lastIndexOf()
   3. includes()
   4. findIndex()

**indexOf()** console.log(fruits.indexOf("Banana"))

1. What does the reverse() method do?
   1. Removes last element
   2. **Reverses the order of array elements**
   3. Sorts alphabetically
   4. Adds to the beginning

**let numbers = [1, 2, 3, 4, 5];**

**numbers.reverse(); // Reverses the order**

**console.log(numbers); // Output: [5, 4, 3, 2, 1]**

1. Which method joins arrays together?
   1. join()
   2. push()
   3. **concat()**
   4. merge()

**let array1 = [1, 2, 3];**

**let array2 = [4, 5, 6];**

**let mergedArray = array1.concat(array2);**

**console.log(mergedArray);**

1. What is the output of [1,2,3].length?
   1. 2
   2. **3**
   3. 4
   4. Error

### Finish This Code (10 marks)

1. Finish this code to add 'banana' to the end of the array: let fruits = ['apple'];

**fruits.push('banana');**

1. Finish this code to remove the first item from the array: let colors = ['red', 'green', 'blue'];

**colors.shift();**

1. Finish this code to replace the second item in the array with 'kiwi': let fruits = ['apple', 'banana', 'cherry'];

**fruits[1] = 'kiwi';**

**fruits.splice(1, 1, 'kiwi'); // Replaces "banana" with "kiwi"**

**console.log(fruits); // Output: ['apple', 'kiwi', 'cherry']**

1. Finish this code to print every item in the array using a for loop: let numbers = [1, 2, 3, 4];

**for (let i = 0; i < numbers.length; i++) {**

**console.log(numbers[i]);**

**}**

**Retrieves the current element at each index**

1. Finish this code to find the last index of 5 in the array: let nums = [1, 5, 3, 5, 2];

**console.log(nums.lastIndexOf(5)); //3**

finds the last occurence

### What Is Wrong With This Code? (12 marks)

1. What is wrong with this code? let fruits = ['apple', 'banana']; fruits.pop('banana');

**The pop() method does not take arguments. It simply removes the last element.**

1. What is wrong wihthis code?

let colors = ['red','green']; console.log(colors[2]);

The array has only **two elements** (0 and 1). Accessing colors[2] returns undefined.

1. What is wrong with this code? let arr = [1, 2, 3];

arr.splice(1, 0, 4);

console.log(arr); // expecting [1, 4, 2, 3]

Correct,Does what its supposed to do.Add,remove,replace.

Start at index 1 which is before the 2.Delete 0 elements,insert the number 4.

# SECTION C OBJECTS

### Multiple Choice Questions (8 marks)

1. Which of the following correctly defines an object?
   1. let person = [name: "John", age: 30];
   2. let person = {name = "John", age = 30};
   3. **let person = {name: "John", age: 30};**
   4. let person = "name: John", "age: 30";
2. Which statement accesses the city property inside a nested object?
   1. person->address->city
   2. **person.address.city**
   3. person.city.address
   4. address.city.person

**let person = {**

**name: "Chantal",**

**age: 25,**

**address: {**

**street: "123 Main St",**

**city: "Cape Town",**

**country: "South Africa"**

**}**

**};**

// Accessing the city property

console.log(person.address.city); // Output: "Cape Town"

1. How do you delete a property from an object?
   1. remove person.age;
   2. **delete person.age;**
   3. person.age = null;
   4. clear person.age

**let person = { name: "Chantal", age: 25 };**

**delete person.age;**

**console.log(person); // Output: { name: "Chantal" }**

1. Which method returns an array of an object's keys?
   1. Object.entries()
   2. Object.values()
   3. **Object.keys()**
   4. Object.listKeys()
2. What is the result of typeof person when person is an object?
   1. **"object"**
   2. "array"
   3. "function"
   4. "string"
3. Which of the following can store methods (functions)?
   1. Arrays only
   2. Strings only
   3. Objects only
   4. **Both arrays and objects**
4. Which notation must be used to access a property with a hyphen in its key?
   1. obj."first-name"
   2. obj.first-name
   3. **obj['first-name']**
   4. obj.first.name
5. What will Object.entries(obj) return?
   1. An array of property names
   2. An array of property values
   3. **An array of [key, value] pairs (**returns an array of key-value pairs from an object.)
   4. An array of methods

Finish This Code (10 marks)

1. Finish this code to log all values in the hobbies array of an object named person. let person = {

name: "Emma",

hobbies: ["painting", "running", "reading"]

};

**console.log(person.hobbies);**

1. Finish this code to add a new property email = ['ali@example.com'](mailto:%27ali@example.com) to the user object. let user = {

name: "Ali"

};

**user.email = 'ali@example.com';**

**console.log(user);**

1. Finish this code to loop through the user object and log all keys and values. let user = {

name: "Sara", age: 22

};

**for (let key in user) {**

**console.log(key + ": " + user[key]);**

**}**

1. Finish this code to define a greet method that logs 'Hello' when called. let person = {

name: "Zack",

// Add greet method/Function here

**greet: function() {**

**console.log("Hello");**

**};**

};

1. Finish this code to access the name of the first user in the array. let users = [

{ name: "Amy" },

{ name: "Leo" }

];

**console.log(users[0].name);**

### What Is Wrong With This Code? (12 marks)

1. What is wrong with this code?

let person = {

name: "Eva" age: 25

};

**The comma , is missing between properties.**

1. What is wrong with this code? let obj = { "first-name" : "Nina" }; console.log(obj.first-name);

**first-name contains a hyphen, so bracket notation must be used**

1. What is wrong with this code?

let data = { skills: ["HTML", "CSS"] }; for (let skill in data.skills) { console.log(skill);

}

This prints **indexes** (0, 1), not actual skills.

**for...in should be used for objects,it will loop through the keys.. but data.skills is an array.** Instead, use:

**for...of is used for arrays, meaning it loops through values (elements).**