Perfect 👍 here are **clear, detailed, exam-oriented answers** for all **5 questions**, with **proper JavaScript code blocks** where required (LPU / Web Technologies style).

---

1. Write a JavaScript code block using arrays and generate the current date in words

To generate the current date in words, JavaScript provides the built-in Date object. Arrays are used to store the **names of days and months**, because the Date object returns numeric values for them.

Steps involved:

1. Create a Date object
2. Use arrays for day names and month names
3. Fetch day, date, month, and year
4. Format the output in words

JavaScript Code:

```
<script>
  var days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];
  var months = ["January", "February", "March", "April", "May", "June",
                "July", "August", "September", "October", "November", "December"];

  var today = new Date();

  var dayName = days[today.getDay()];
  var monthName = months[today.getMonth()];
  var date = today.getDate();
  var year = today.getFullYear();

  if (date < 10) {
    date = "0" + date;
  }

  var result = dayName + ", " + monthName + " " + date + ", " + year;
  document.write(result);
</script>
```

Output format:

Saturday, October 09, 2021

Explanation:

- getDay() → returns day index (0–6)
- getMonth() → returns month index (0–11)
- Arrays convert numeric values into readable words

---

2. What if I put a function into my script but decide not to call it? Will it matter?

If a function is written in a JavaScript script but **never called**, it will **not execute** and **will not affect the program output**.

Explanation:

- JavaScript loads the function into memory
- The function runs **only when it is called**
- Uncalled functions do not cause errors

Example:

```
function greet() {
  alert("Hello!");
}
// Function is never called
```

Result:

- No alert is shown
- No error occurs

Conclusion:

It does **not matter functionally**, but unused functions should be removed to keep code clean and efficient.

---

3. What happens if I decide to remove a function from my script later?

Removing a function from a script **will not cause any problem** as long as that function is **not being used anywhere else**.

Scenarios:

Case 1: Function is not used

- Safe to remove
- No effect on program

Case 2: Function is used somewhere

- JavaScript will throw an error
  Example:

Uncaught ReferenceError: functionName is not defined

Best practice:

- Always search where the function is called before removing it
- Remove unused code to improve performance and readability

---

4. What is the best way to determine when to use a function and when to code directly in the script?

Choosing between using a function or writing direct code depends on **reusability, clarity, and complexity**.

Use a function when:

- Code is repeated multiple times
- Logic is complex
- Code needs to be reused
- Program readability is important

Example:

```
function calculateSum(a, b) {
  return a + b;
}
```

Write direct code when:

- Logic is simple
- Code is used only once
- Script is very small

Example:

```
var sum = 5 + 10;
```

Conclusion:

Functions make code **modular, reusable, and maintainable**, which is essential for large applications.

---

5. Write a JavaScript function which accepts an argument and returns the type

JavaScript provides the typeof operator to determine the **data type of a variable**.

Possible return values of typeof:

- object
- boolean
- function
- number
- string
- undefined

JavaScript Function:

```
function getType(value) {
  return typeof value;
}
```

Example Usage:

```
getType(10);           // "number"
getType("Hello");      // "string"
getType(true);         // "boolean"
getType({});           // "object"
getType(function(){}); // "function"
getType(x);            // "undefined"
```

Explanation:

- Function accepts one argument
- Uses typeof operator
- Returns the data type as a string
