JavaScript:

JavaScript is a high-level, flexible, object oriented and interpreted programming language mainly used for web development. It enables developers to add dynamic and interactive elements to websites. JavaScript is a key component of the modern web environment and is supported by most web browsers. JavaScript is commonly used in conjunction with HTML and CSS to build interactive and dynamic websites. Over the years, various libraries, and frameworks, such as jQuery, React, Angular, and Vue.js, have been developed to simplify and increase the development process using JavaScript.

Key features of JavaScript include:

Client-Side Scripting: JavaScript is mainly used for client-side scripting, allowing developers to create dynamic content, manipulate the Document Object Model (DOM), and response to user interactions within a web browser.

Object-Oriented: JavaScript is an object-oriented language, supporting the creation and manipulation of objects. Objects can encapsulate data which helps in creating reuseable module.

Event-Driven: JavaScript is event-driven, meaning it can respond to user actions such as clicks, keyboard inputs, and other events. This makes it well-suited for creating interactive and responsive web applications.

Asynchronous Programming: JavaScript supports asynchronous programming through features like callbacks, promises, and async/await. This allows handling tasks like fetching data from servers without blocking the execution of other code.

Cross-Browser Compatibility: JavaScript is supported by all major web browsers which make it a universal language for client-side web development.

Server-Side Development: With the initiation of technologies like *Node.js*, JavaScript can also be used for server-side development, allowing developers to use the same language on both the client and server sides of a web application.

Uses of JavaScript:

- Client-side validation
- Dynamic drop-down menus.
- Displaying date and time in web-based applications
- Display pup-up windows and dialog boxes.

Adding JavaScript to HTML:

- Embodying the JavaScript code between <script> and </script>
- Creating external JavaScript file with .js extension and load/link in HTML document.
- Placing JavaScript code inline directly inside an HTML tag using special tag attributes.

Embedding the JavaScript code:

We can place JavaScript code directly by placing between <script> and </script> tags. The <script> tag indicates that the contained statements are to be interpreted as executable script, not HTML.

Example

```
<!DOCTYPE HTML>
  <html>
  <head>
            <title> Embedding JavaScript</title>
  </head>
  <body>
  <script>
            var hi= ‘Hello World!!!’;
            document.write(hi);
  <script>
  <body>
  </html>
```

Creating external JavaScript:

We can create external .js file and with the help of src attributes of <script> tag that file can be called. Example: <script src= “js\hi.js”></script>. To call external JS file curtain event should occurred in the webpage.

Placing JavaScript code inline:

Inline JS can be place directly inside HTML tag using the special tag event like onclick, on mouseover, on keypress, onload etc. However, we should avoid placing large amount of JS code inline.

Example

```
<! DOCTYPE HTML>
<html>
        <head>
                <title> Inline JavaScript</title>
        </head>
        <body>
                <button onclick= “alert(‘Hello
                world’)”> Click Me<button>
        <body>
</html>
```

JavaScript Fundamental:

- Variables
  - var, let key are used.
  - const key word is used to declared constant value.
  - Types: Local, Global.
- Data types:
  - Primitive data type
  - Non- primitive data type
- Arithmetic, Assignment, and comparison Operators.
- Unary, Logical, Comma, and spread Operators.
- Functions

Object-Based Programming languages:

Object based programming languages is also known as Object Oriented Programming (OOP) Language. OOP is the style of programming characterized by the identification of class of object closely linked with the methods(functions) with which they are associated. It also includes ideas of inheritance of attributes and methods. The four principles of OOP are encapsulation, abstraction, inheritance, and polymorphism.

JavaScript is a templet based not class-based OOP language. Here, we do not crate a class to get the object, but objects are directly created. There are three ways to create a object in JavaScript. They are:

1. **By object literal:** The syntax of creating a new object literal is given below:

   object={property1:value1, property2:value2, property3:value3,………., propertyN:valueN}

2. **By creating an instance of object directly (using the new keyword):** The syntax of creating object directly is given as: var objectname= new object(); Here, a new key word is used to create an object.
3. **By using an object constructor (using the new keyword):** Functions are created with arguments. Each argument value can be assigned to current object by using the keyword. This keyword refers to the current object. Some popular predefined object of JavaScript are as follows:
   1. **Date Object:** The Date object is a datatype built into the JavaScript language. Date object are created as: new date( ).
   2. **Number Object:** The Number object represents the numerical date, either integers or floating-point numbers. Syntax: var val= new Number(number);
   3. **String Object:** The String object are JavaScript objects that represents a sequence of characters. In JavaScript string can be created in two ways. They are given below.
      a. **By String literal:** The string literal are created using double quotes.
         Syntax: var strName= “String Value”.
      b. **By String object (use new):** Here new key word is used to crate an instance of string only.
         Syntax: var strName= new String (String literal).

4. **Array Object:** In JavaScript array is an object that represents a collection of similar or different type of elements. Syntax: var arrayName= new Array (“Value1”, (“Value1”, “Value2”, ……..,“ValueN”);
5. Maths Object: The maths object provides the properties and methods for mathematical constant and functions. Syntax: var val=Math.Math_Object;
   Example: var sine_val=Math.sin(30);

Some Important Questions

1. Write Short note on internet technology.
2. Different between server-side and client-side scripting languages.
3. Explain different types of data types used in JavaScript.
4. What are the uses of JavaScript in web development?
5. What is object-based programming? Explain the object of JavaScript.
6. Write a JavaScript code to calculate the factorial of given number.
7. Write a function to enter two number and find out the sum of two numbers in JavaScript.