Tech Mahindra 2024 Interview questios-hiringhustle

Tech Mahindhra common interview questions for JavaScript along with simple answers:

1. *What is JavaScript?*
    - JavaScript is a high-level, interpreted programming language primarily used for adding interactivity and dynamic behavior to web pages.
2. *What are the different data types in JavaScript?*
    - JavaScript has six primitive data types: string, number, boolean, null, undefined, and symbol. Additionally, there is one complex data type, which is object.
3. *What is the difference between null and undefined?*
    - null represents the intentional absence of any value, whereas undefined typically means a variable has been declared but not assigned a value.
4. *How do you declare variables in JavaScript?*
    - Variables in JavaScript can be declared using the var, let, or const keywords. For example:
      javascript
      var x = 5;
      let y = 'hello';
      const PI = 3.14;
5. *What is the difference between var, let, and const?*
    - var has function scope, let has block scope, and const also has block scope but cannot be reassigned once declared.
6. *What is a closure in JavaScript?*
    - A closure is a function that has access to its own scope, the outer function's scope, and the global scope, even after the outer function has finished executing.

7. *Explain event bubbling and event capturing.*
    - Event bubbling refers to the event propagation mechanism where the event starts at the innermost target element and moves outward to the outermost ancestor. Event capturing, on the other hand, starts from the outermost element and moves towards the target element.
8. *What is the this keyword in JavaScript?*
    - The this keyword refers to the object that is currently executing the JavaScript code. Its value is determined by how a function is called.
9. *What is a callback function?*
    - A callback function is a function that is passed as an argument to another function and is executed after the completion of a specific task.
10. *What are arrow functions in JavaScript?*
    - Arrow functions are a concise way to write function expressions in JavaScript. They use a shorter syntax compared to traditional function expressions and do not bind their own this value.

Asked for Both Freshers and Exp

1. *Explain the concept of hoisting in JavaScript.*
    - Hoisting is JavaScript's default behavior of moving variable and function declarations to the top of their containing scope during the compile phase.
2. *What is the difference between == and === in JavaScript?*
    - == checks for equality after type conversion, while === checks for equality without type conversion (strict equality).
3. *What are prototypes in JavaScript?*
    - Prototypes are a mechanism by which JavaScript objects inherit properties and methods from other objects. Every JavaScript object has a prototype property, which makes inheritance possible.
4. *What is event delegation in JavaScript?*
    - Event delegation is a technique where a single event listener is attached to a parent element, rather than multiple event listeners being attached to individual child elements. This can improve performance and simplify code.

5. *Explain the concept of asynchronous programming in JavaScript.*
    - Asynchronous programming allows multiple operations to be performed concurrently, without blocking the execution of subsequent code. This is commonly achieved using callbacks, promises, or async/await.
6. *What are promises in JavaScript?*
    - Promises are objects representing the eventual completion or failure of an asynchronous operation. They allow you to handle asynchronous operations more easily and cleanly, avoiding callback hell.
7. *What is the event loop in JavaScript?*
    - The event loop is the mechanism that allows JavaScript to perform non-blocking I/O operations, such as asynchronous callbacks, while still maintaining responsiveness and handling events.
8. *How do you handle errors in JavaScript?*
    - Errors in JavaScript can be handled using try-catch blocks, where code that might throw an error is placed inside a try block and any errors are caught and handled in a catch block.
9. *Explain the concept of scope in JavaScript.*
    - Scope refers to the visibility and accessibility of variables within a JavaScript program. JavaScript has function scope, meaning variables are accessible only within the function in which they are declared, and block scope introduced with let and const.
10. *What is the difference between synchronous and asynchronous code execution in JavaScript?*
    - Synchronous code executes in sequence, blocking further execution until the current operation is completed. Asynchronous code allows multiple operations to be performed concurrently, without blocking the execution of subsequent code.

Python Tech Mahindra Interview Question in 2024-2023

List: Lists are mutable, meaning you can add, remove, or modify elements after the list has been created. They are created using square brackets [].

Tuple: Tuples are immutable, meaning once they are created, their elements cannot be changed or modified. They are created using parentheses ().

Here's a summary of the differences:

Lists are typically used for collections of similar items where the order and number of elements might change.
Tuples are often used for fixed collections of items, such as coordinates or settings, where the elements should not be changed

- `range()`: In Python 2, `range()` returns a list containing the specified range of numbers. It generates all the numbers in the sequence and stores them in memory, which can be memory-intensive for large ranges.
- `xrange()`: In Python 2, `xrange()` generates the numbers in the sequence on-the-fly, without storing them all in memory. It returns an `xrange` object, which behaves like an iterator, generating each number as needed. This makes `xrange()` more memory-efficient, especially for large ranges.

In Python 3, `range()` behaves like Python 2's `xrange()`, returning a `range` object that generates numbers on-the-fly, while `xrange()` is no longer available.

Pickling is the process of converting a Python object into a byte stream, which can be stored in a file or transmitted over a network.
This byte stream can then be saved to a file using the pickle.dump() function or sent over a network connection.
Pickling allows you to preserve the state of an object, including its data and internal state, so that it can be reconstructed later.

Unpickling is the process of converting a byte stream back into a Python object.

This is done using the pickle.load() function, which reads the byte stream from a file or network connection and reconstructs the original object.
Unpickling allows you to restore the state of an object from a previously pickled byte stream.

In Python, there are several built-in data types that are commonly used to store and manipulate data. Here are some of the main data types:

1. *Numeric Types*:
    - int: Integer numbers, e.g., 5, -3, 0.
    - float: Floating-point numbers, e.g., 3.14, -0.5, 2.0.
2. *Sequence Types*:
    - list: Ordered collection of items, mutable (modifiable), e.g., [1, 2, 3].
    - tuple: Ordered collection of items, immutable (cannot be modified), e.g., (1, 2, 3).
    - range: Represents a sequence of numbers, often used for looping, e.g., range(5) produces 0, 1, 2, 3, 4.
3. *Mapping Type*:
    - dict: Collection of key-value pairs, unordered and mutable, e.g., {'name': 'John', 'age': 30}.
4. *Set Types*:
    - set: Unordered collection of unique items, mutable, e.g., {1, 2, 3}.
    - frozenset: Immutable set, e.g., frozenset({1, 2, 3}).
5. *Boolean Type*:
    - bool: Represents boolean values True and False.
6. *Text Type*:
    - str: Represents strings of characters, e.g., "hello", 'world'.
7. *Binary Types*:
    - bytes: Represents sequences of bytes, immutable, e.g., b'hello'.
    - bytearray: Mutable sequence of bytes, e.g., bytearray(b'hello').

- memoryview: Provides access to the internal data of objects that support the buffer protocol.

These are the main built-in data types in Python. They provide a wide range of options for storing and manipulating different kinds of data in Python programs.

In Python, scope refers to the visibility of variables. Variables are only accessible within the region they are defined, which is called their scope. There are three types of scopes:

- Global Scope: If a variable is declared outside all the functions, it is available to all the functions in the program. This is called a global variable.
- Local Scope: If a variable is declared inside a function, it is only available within that function. This is called a local variable.
- Nonlocal Scope: Nonlocal variables are used in nested functions. If we need to modify a variable that is outside of the immediate scope, we can declare it with the 'nonlocal' keyword.

Here is the Python code for finding the maximum number in an array:

```
def max_arr(arr):
    max = float('-inf')
    for i in arr:
        if i > max:
            max = i
    return max

arr = [1, 2, 3, 4, 5]
print(max_arr(arr))
```

This function works by initially setting the maximum to negative infinity, then iterating over each element in the array. If an element is larger than the current maximum, it becomes the new maximum. Finally, the maximum is returned.

Here is a Python function that reverses a number:

```
def reverse_number(number):
    reversed_number = 0
```

```
    while number != 0:
        digit = number % 10  # Get the last digit
        reversed_number = reversed_number * 10 + digit  # A
ppend the digit to the reversed number
        number //= 10  # Move to the next digit by removing
the last digit
    return reversed_number
```

You can use this function to reverse a number like this:

```
number = 12345
print(reverse_number(number))  # Output: 54321
```

Here is a Python function to reverse a string:

```
def reverse_string(s):
    return s[::-1]

s = "Hello, World!"
print(reverse_string(s))  # Output: "!dlroW ,olleH"
```

This function uses Python's slice syntax to reverse the string. The [::-1] slice means start at the end of the string and end at position 0, move with the step -1, negative one, which means one step backwards.

Here is a Python function to calculate the factorial of a number:

```
def factorial(n):
    fact = 1
    for i in range(1, n + 1):
        fact *= i
    return fact

number = 5
print(factorial(number))  # Output: 120
```

This function works by looping from 1 to the input number, and multiplying each number by the current factorial value. The resulting factorial value is then returned.

Here is the Python code for checking whether a number is a palindrome:

```
def is_palindrome(n):
    temp = n
    rev = 0
    while(n > 0):
        digit = n % 10
        rev = rev * 10 + digit
        n = n // 10
    if(temp == rev):
        return True
    else:
        return False

number = 121
if is_palindrome(number):
    print("The number is a palindrome!")
else:
    print("The number isn't a palindrome!")
```

This function works by reversing the input number and then comparing the reversed number with the original number. If they are equal, the number is a palindrome.

Here is the Python code for generating Fibonacci series:

```
def fibonacci(n):
    if n <= 0:
        return "Incorrect input"
    elif n == 1:
        return 0
    elif n == 2:
        return 1
```

```
    else:
        a, b = 0, 1
        for _ in range(2, n):
            a, b = b, a + b
        return b

print(fibonacci(9))
```

This function uses an iterative approach to generate the Fibonacci series. It starts with two variables `a` and `b` initialized to `0` and `1` respectively. Then in a for loop, it calculates the nth Fibonacci number by adding the last two numbers in the series.

Object-Oriented Programming (OOP) is a programming paradigm that relies on the concept of "classes" and "objects". It is used to structure a software program into simple, reusable pieces of code blueprints (usually called classes), which are used to create individual instances of objects.

There are four fundamental concepts in OOP:

1. **Inheritance**: This is a way of creating a new class using properties and methods of an existing class while adding some more features. It provides code reusability, makes it easier to create and maintain an application. The new class is known as derived class or child class, and the one whose properties are inherited is known as the base class or parent class.
2. **Encapsulation**: It is a way of wrapping data and the methods that work on data within one unit. This puts restrictions on accessing variables and methods directly and can prevent the accidental modification of data. To access the data, we use public methods (getters and setters) which ensure the secure handling of data.
3. **Polymorphism**: It allows one interface to be used for a general class of actions. The specific action is determined by the exact nature of the situation. It provides a way to use a class exactly like its parent so there's no confusion with mixing types. But each child class keeps its own methods as they are.

4. **Abstraction**: It is a process of hiding the implementation details and showing only functionality to the user. In other words, it deals with the outside view of an object (interface). The user will have the information on what the object does instead of how it does it.

Python provides all the standard features of object-oriented programming. The class is the blueprint that defines a nature of a future object. An instance is a specific object created from a particular class.

Tech Mahindra Java Interview Questions:

1. *What is Java?*
    - Java is a high-level, object-oriented programming language developed by Sun Microsystems. It is platform-independent, secure, and robust, with a syntax similar to C and C++.
2. *Explain the basic structure of a Java program.*
    - A Java program begins with a package declaration, followed by import statements, class declaration, main method declaration, and statements inside the main method.
3. *What is JVM and its role in Java?*
    - JVM stands for Java Virtual Machine. It is the component of the Java runtime environment that executes Java bytecode. It provides a platform-independent way of executing Java code, as it is responsible for converting the bytecode into machine code.
4. *What are the principles of Object-Oriented Programming (OOP)?*
    - The principles of OOP are encapsulation, inheritance, polymorphism, and abstraction.
5. *What is the difference between an interface and an abstract class in Java?*
    - An abstract class can have both abstract and non-abstract methods, while an interface can have only abstract methods. Also, a class can implement multiple interfaces but can only inherit from one abstract class.
6. *Explain the concept of exception handling in Java.*

- Exception handling in Java is a powerful mechanism to handle runtime errors so that the normal flow of the application can be maintained.

7. *What is multithreading in Java and how is it achieved?*
    - Multithreading is a feature in Java that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. It can be achieved by implementing the Runnable interface or extending the Thread class.
8. *What is the difference between ArrayList and LinkedList in Java?*
    - The main difference lies in their implementation which causes different performance for different operations. ArrayList is implemented as a resizable array, while LinkedList is implemented as a doubly linked list.
9. *What is the difference between equals() method and == operator?*
    - The == operator checks if two references point to the same object in memory, whereas the equals() method can be overridden to check if two objects have the same logical value.
10. *What are Java Annotations?*
    - Annotations are tags that associate metadata with program elements, such as classes, methods, variables, parameters, and packages.

Asked for Both Freshers and Experienced

1. *What is the use of a final keyword in Java?*
    - The final keyword in Java is used to restrict the user. A final variable can't be modified, a final method can't be overridden and a final class can't be inherited.
2. *What is JDBC?*
    - JDBC stands for Java Database Connectivity. It is a standard Java API for database-independent connectivity between Java & a wide range of databases.
3. *What is the difference between throw and throws in Java?*
    - The throw keyword is used to explicitly throw an exception from a method or any block of code, while throws is used to declare an exception and delay handling it to the caller of the method.
4. *What do you understand by Java Servlets?*

- Java Servlets are server-side Java programming elements which are used to create web applications. It provides a component-based, platform-independent method for building Web-based applications.

5. *What is the difference between a JDK and a JVM?*
    - JDK stands for Java Development Kit. It is a software development environment used for developing Java applications. It includes the JVM, Java libraries, and other modules to complete the development of a Java Application. JVM stands for Java Virtual Machine. It is the platform-independent virtual machine that executes the Java code.
6. *Explain garbage collection in Java.*
    - Garbage collection in Java is a process by which the programs perform automatic memory management. Java programs compile into bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program.
7. *What is Java Reflection API?*
    - Java Reflection API is a powerful feature, used to inspect, analyze, and modify the runtime behavior of class, interface, enum, and method at runtime.
8. *What is the difference between fail-fast and fail-safe in Java?*
    - Fail-fast iterators throw a ConcurrentModificationException, if a collection is modified while iterating over it. On the other hand, fail-safe iterators don't throw any exception if a collection is modified while iterating over it because they work on a clone of the collection instead of the original collection.
9. *What is method overloading and method overriding in Java?*
    - Method overloading is a feature in Java that allows a class to have two or more methods having the same name, if their argument lists are different. Method overriding is a feature that allows a subclass to provide a specific implementation of a method that is already provided by its parent class.
10. *What is the difference between a static and a non-static method in Java?*
    - A static method belongs to the class rather than the object. It can be called without creating an instance of the class. A non-static method is

specific to an object of a class and can manipulate the instance variables of the object.

Here are some coding questions that could be asked during a Tech Mahindra Java interview:

1. *Write a Java program to reverse a string without using inbuilt String API.*

```
public class Main {
    public static void main(String[] args) {
        String str = "Tech Mahindra";
        String reversed = reverseString(str);
        System.out.println("Reversed String: " + reversed);
    }

    public static String reverseString(String str) {
        char[] chars = str.toCharArray();
        int left = 0;
        int right = str.length() - 1;
        while (left < right) {
            char temp = chars[left];
            chars[left] = chars[right];
            chars[right] = temp;
            left++;
            right--;
        }
        return new String(chars);
    }
}
```

1. *Write a Java program to find the second highest number in an array.*

```
public class Main {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5};
        System.out.println("Second highest: " + secondHighe
st(numbers));
    }
```

```
    public static int secondHighest(int[] numbers) {
        int highest = Integer.MIN_VALUE;
        int secondHighest = Integer.MIN_VALUE;

        for (int number : numbers) {
            if (number > highest) {
                secondHighest = highest;
                highest = number;
            } else if (number > secondHighest && number < h
ighest) {
                secondHighest = number;
            }
        }

        return secondHighest;
    }
}
```

1. *Write a Java program to check whether a number is prime or not.*

```
public class Main {
    public static void main(String[] args) {
        int num = 29;
        boolean flag = false;

        for (int i = 2; i <= num / 2; ++i) {
            if (num % i == 0) {
                flag = true;
                break;
            }
        }

        if (!flag)
            System.out.println(num + " is a prime numbe
r.");
        else
            System.out.println(num + " is not a prime numbe
```

```
r.");
    }
}
```

Remember, these are just examples. The actual questions can vary and might be more complex.

Tech Mahindra SQL Interview Questions:

1. *What is SQL?*
    - SQL stands for Structured Query Language. It is a standard language used to interact with relational databases, including creating, deleting, fetching rows, and modifying rows data.
2. *What are the different types of SQL commands?*
    - The types of SQL commands are Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and Transaction Control Language (TCL).
3. *What is a primary key?*
    - A primary key is a column or a set of columns that uniquely identifies each row in the table. It must contain unique values and cannot contain null values.
4. *What is a foreign key?*
    - A foreign key is a column or a set of columns used to establish a link between the data in two tables.
5. *What is a join in SQL?*
    - A JOIN clause in SQL is used to combine rows from two or more tables, based on a related column between them.
6. *What are the different types of joins in SQL?*
    - The different types of joins in SQL are INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
7. *What is a NULL value in SQL?*
    - A NULL value in SQL is used to represent missing or unknown data. It is different from zero or an empty string.

8. *What is normalization?*
    - Normalization is a process in database design that organizes tables to reduce redundancy and dependency of data.
9. *What is a SQL View?*
    - A View in SQL is a virtual table based on the result-set of an SQL statement. It contains rows and columns, just like a real table.
10. *What is a stored procedure?*
    - A stored procedure is a prepared SQL code that can be saved and reused. It can be used to perform actions that would require multiple lines of code.
11. *What is the difference between DELETE and TRUNCATE commands?*
    - DELETE is a DML command and is used to delete a row in a table. TRUNCATE is a DDL command and is used to delete all the rows from a table.
12. *What is SQL injection?*
    - SQL injection is a code injection technique that might destroy your database. It is one of the most common web hacking techniques.
13. *What are SQL Constraints?*
    - SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table.
14. *What is a Database Cursor?*
    - Database Cursor is a control structure that allows for traversal of records in a database.
15. *What are the differences between clustered and non-clustered indexes?*

- The differences between clustered and non-clustered indexes in SQL are based on storage of data in the database, the creation of index tables, and level of speed in fetching data from the database.

1. **Query to select all records from a table**

```
SELECT * FROM table_name;
```

1. **Query to select specific columns from a table**

```
SELECT column1, column2 FROM table_name;
```

1. **Query to sort records in descending order**

```
SELECT * FROM table_name ORDER BY column_name DESC;
```

1. **Query to retrieve unique values in a column**

```
SELECT DISTINCT column_name FROM table_name;
```

1. **Query to count the number of records in a table**

```
SELECT COUNT(*) FROM table_name;
```

1. **Query to retrieve records with a condition**

```
SELECT * FROM table_name WHERE condition;
```

1. **Query to use aggregate functions like `SUM`, `AVG`, `MAX`, `MIN`**

```
SELECT SUM(column_name) FROM table_name WHERE condition;
```

1. **Query to group records and apply count on each group**

```
SELECT column1, COUNT(column2) FROM table_name GROUP BY col
umn1;
```

1. **Query to use INNER JOIN to get records from multiple tables**

```
SELECT * FROM table1 INNER JOIN table2 ON table1.matching_c
olumn = table2.matching_column;
```

1. **Query to update records in a table**

```
UPDATE table_name SET column1 = value1, column2 = value2 WH
ERE condition;
```

1. **Query to delete records from a table**

```
DELETE FROM table_name WHERE condition;
```

Remember, these are just basic SQL queries. Depending on the complexity of the database and the specific requirements, the queries could be much more complex.