// JavaScript Assignment Questions

// 1. W.A.P that uses a for loop to print the numbers from 1 to 10.

// 2. W.A.P that uses a for loop to print the numbers from 10 to 1.

// 3. W.A.P that prints all even numbers between 1 and 20 using a for loop.

// 4. W.A.P that prints the sum of numbers from 1 to 10 using a for loop.

// 5. W.A.P that prints the prime numbers 1 to 20 using a for loop.

// 6. W.A.P that uses a while loop to print the numbers from 1 to 10.

// 7. W.A.P that uses a while loop to print the numbers from 10 to 1.

// 8. W.A.P that prints all even numbers between 1 and 20 using a while loop.

// 9. W.A.P that prints the sum of numbers from 1 to 10 using a while loop.

// 10. W.A.P that prints the prime numbers 1 to 20 using a while loop.

// 11. W.A.P that uses a do-while loop to print the numbers from 1 to 10.

// 12. W.A.P that uses a do-while loop to print the numbers from 10 to 1.

// 13. W.A.P that prints all even numbers between 1 and 20 using a do while loop.

// 14. W.A.P that prints the sum of numbers from 1 to 10 using a do while loop.

// 15. W.A.P that prints the prime numbers 1 to 20 using a do while loop.

// 1. W.A.P that uses a for loop to print the numbers from 1 to 10.

// JavaScript program to print numbers from 1 to 10 using a for loop

// Using a for loop

for (*var* i = 1; i <= 10; i++) {

    console.log(i);

}

// Tracking:

// Initializing i to 1

i = 1;

// Checking the condition: 1 <= 10 (true)

// Entering the loop

// Output: 1

i++;

// Checking the condition: 2 <= 10 (true)

// Entering the loop

// Output: 2

i++;

// Checking the condition: 3 <= 10 (true)

// Entering the loop

// Output: 3

i++;

// Checking the condition: 4 <= 10 (true)

// Entering the loop

// Output: 4

i++;

// Checking the condition: 5 <= 10 (true)

// Entering the loop

// Output: 5

i++;

// Checking the condition: 6 <= 10 (true)

// Entering the loop

// Output: 6

i++;

// Checking the condition: 7 <= 10 (true)

// Entering the loop

// Output: 7

i++;

// Checking the condition: 8 <= 10 (true)

// Entering the loop

// Output: 8

i++;

// Checking the condition: 9 <= 10 (true)

// Entering the loop

// Output: 9

i++;

// Checking the condition: 10 <= 10 (true)

// Entering the loop

// Output: 10

i++;

// Checking the condition: 11 <= 10 (false)

// Condition is false, exiting the loop

//So, the output of the code would be:

1

2

3

4

5

6

7

8

9

10

// 2. W.A.P that uses a for loop to print the numbers from 10 to 1.

// JavaScript program to print numbers from 10 to 1 using a for loop

// Using a for loop

for (*var* i = 10; i >= 1; i--) {

    console.log(i);

}

//Tracking:

// Initializing i to 10

i = 10;

// Checking the condition: 10 >= 1 (true)

// Entering the loop

// Output: 10

i--;

// Checking the condition: 9 >= 1 (true)

// Entering the loop

// Output: 9

i--;

// Checking the condition: 8 >= 1 (true)

// Entering the loop

// Output: 8

i--;

// Checking the condition: 7 >= 1 (true)

// Entering the loop

// Output: 7

i--;

// Checking the condition: 6 >= 1 (true)

// Entering the loop

// Output: 6

i--;

// Checking the condition: 5 >= 1 (true)

// Entering the loop

// Output: 5

i--;

// Checking the condition: 4 >= 1 (true)

// Entering the loop

// Output: 4

i--;

// Checking the condition: 3 >= 1 (true)

// Entering the loop

// Output: 3

i--;

// Checking the condition: 2 >= 1 (true)

// Entering the loop

// Output: 2

i--;

// Checking the condition: 1 >= 1 (true)

// Entering the loop

// Output: 1

i--;

// Checking the condition: 0 >= 1 (false)

// Condition is false, exiting the loop

//So, the output of the code would be:

10

9

8

7

6

5

4

3

2

1

3. W.A.P that prints all even numbers between 1 and 20 using a for loop.

// 3 JavaScript program to print even numbers between 1 and 20 using a for loop

// Using a for loop

for (*var* i = 2; i <= 20; i += 2) {

  console.log(i);

}

//Tracking:

// Initializing i to 2

i = 2;

// Checking the condition: 2 <= 20 (true)

// Entering the loop

// Output: 2

i += 2;

// Checking the condition: 4 <= 20 (true)

// Entering the loop

// Output: 4

i += 2;

// Checking the condition: 6 <= 20 (true)

// Entering the loop

// Output: 6

i += 2;

// Checking the condition: 8 <= 20 (true)

// Entering the loop

// Output: 8

i += 2;

// Checking the condition: 10 <= 20 (true)

// Entering the loop

// Output: 10

i += 2;

// Checking the condition: 12 <= 20 (true)

// Entering the loop

// Output: 12

i += 2;

// Checking the condition: 14 <= 20 (true)

// Entering the loop

// Output: 14

i += 2;

// Checking the condition: 16 <= 20 (true)

// Entering the loop

// Output: 16

i += 2;

// Checking the condition: 18 <= 20 (true)

// Entering the loop

// Output: 18

i += 2;

// Checking the condition: 20 <= 20 (true)

// Entering the loop

// Output: 20

i += 2;

// Checking the condition: 22 <= 20 (false)

// Condition is false, exiting the loop

//So, the output of the code would be:

2

4

6

8

10

12

14

16

18

20

4. W.A.P that prints the sum of numbers from 1 to 10 using a for loop.

// 4 JavaScript program to print the sum of numbers from 1 to 10 using a for loop

// Initializing a variable to store the sum

*var* sum = 0;

// Using a for loop to iterate from 1 to 10

for (*var* i = 1; i <= 10; i++) {

    sum += i; // Adding the current value of i to the sum

}

// Printing the sum to the console

console.log("The sum of numbers from 1 to 10 is: " + sum);

//Tracking:

// Initializing i to 1

i = 1;

// Checking the condition: 1 <= 10 (true)

// Entering the loop

// Adding 1 to the sum: 0 + 1 = 1

i++;

// Checking the condition: 2 <= 10 (true)

// Entering the loop

// Adding 2 to the sum: 1 + 2 = 3

i++;

// Checking the condition: 3 <= 10 (true)

// Entering the loop

// Adding 3 to the sum: 3 + 3 = 6

i++;

// Checking the condition: 4 <= 10 (true)

// Entering the loop

// Adding 4 to the sum: 6 + 4 = 10

i++;

// Checking the condition: 5 <= 10 (true)

// Entering the loop

// Adding 5 to the sum: 10 + 5 = 15

i++;

// Checking the condition: 6 <= 10 (true)

// Entering the loop

// Adding 6 to the sum: 15 + 6 = 21

i++;

// Checking the condition: 7 <= 10 (true)

// Entering the loop

// Adding 7 to the sum: 21 + 7 = 28

i++;

// Checking the condition: 8 <= 10 (true)

// Entering the loop

// Adding 8 to the sum: 28 + 8 = 36

i++;

// Checking the condition: 9 <= 10 (true)

// Entering the loop

// Adding 9 to the sum: 36 + 9 = 45

i++;

// Checking the condition: 10 <= 10 (true)

// Entering the loop

// Adding 10 to the sum: 45 + 10 = 55

i++;

// Checking the condition: 11 <= 10 (false)

// Condition is false, exiting the loop

//So, the output of the code would be:

The sum of numbers from 1 to 10 is: 55

5. W.A.P that prints the prime numbers 1 to 20 using a for loop.

// 5 JavaScript program to print prime numbers between 1 and 20 using a for loop

// Function to check if a number is prime

*function* isPrime(*num*) {

    if (*num* <= 1) return false;

    for (*var* i = 2; i <= Math.sqrt(*num*); i++) {

        if (*num* % i === 0) {

            return false;

        }

    }

    return true;

}

// Using a for loop to iterate from 1 to 20

for (*var* i = 1; i <= 20; i++) {

    if (isPrime(i)) {

        console.log(i);

    }

}

// Initializing i to 1

i = 1;

// Checking the condition: 1 <= 20 (true)

// Entering the loop

// Calling isPrime(1), which returns false as 1 is not a prime number

// No output for 1

i++;

// Checking the condition: 2 <= 20 (true)

// Entering the loop

// Calling isPrime(2), which returns true as 2 is a prime number

// Output: 2

i++;

// Checking the condition: 3 <= 20 (true)

// Entering the loop

// Calling isPrime(3), which returns true as 3 is a prime number

// Output: 3

i++;

// Checking the condition: 4 <= 20 (true)

// Entering the loop

// Calling isPrime(4), which returns false as 4 is not a prime number

// No output for 4

i++;

// Checking the condition: 5 <= 20 (true)

// Entering the loop

// Calling isPrime(5), which returns true as 5 is a prime number

// Output: 5

i++;

// Checking the condition: 6 <= 20 (true)

// Entering the loop

// Calling isPrime(6), which returns false as 6 is not a prime number

// No output for 6

i++;

// Checking the condition: 7 <= 20 (true)

// Entering the loop

// Calling isPrime(7), which returns true as 7 is a prime number

// Output: 7

i++;

// Checking the condition: 8 <= 20 (true)

// Entering the loop

// Calling isPrime(8), which returns false as 8 is not a prime number

// No output for 8

i++;

// Checking the condition: 9 <= 20 (true)

// Entering the loop

// Calling isPrime(9), which returns false as 9 is not a prime number

// No output for 9

i++;

// Checking the condition: 10 <= 20 (true)

// Entering the loop

// Calling isPrime(10), which returns false as 10 is not a prime number

// No output for 10

i++;

// Checking the condition: 11 <= 20 (true)

// Entering the loop

// Calling isPrime(11), which returns true as 11 is a prime number

// Output: 11

i++;

// Checking the condition: 13 <= 20 (true)

// Entering the loop

// Calling isPrime(12), which returns false as 12 is not a prime number

// No output for 12

i++;

// Checking the condition: 13 <= 20 (true)

// Entering the loop

// Calling isPrime(13), which returns true as 13 is a prime number

// Output: 13

i++;

// Checking the condition: 14 <= 20 (true)

// Entering the loop

// Calling isPrime(14), which returns false as 14 is not a prime number

// No output for 14

i++;

// Checking the condition: 15 <= 20 (true)

// Entering the loop

// Calling isPrime(15), which returns false as 15 is not a prime number

// No output for 15

i++;

// Checking the condition: 16 <= 20 (true)

// Entering the loop

// Calling isPrime(16), which returns false as 16 is not a prime number

// No output for 16

i++;

// Checking the condition: 17 <= 20 (true)

// Entering the loop

// Calling isPrime(17), which returns true as 17 is a prime number

// Output: 17

i++;

// Checking the condition: 18 <= 20 (true)

// Entering the loop

// Calling isPrime(18), which returns false as 18 is not a prime number

// No output for 18

i++;

// Checking the condition: 19 <= 20 (true)

// Entering the loop

// Calling isPrime(19), which returns true as 19 is a prime number

// Output: 19

i++;

// Checking the condition: 20 <= 20 (true)

// Entering the loop

// Calling isPrime(20), which returns false as 20 is not a prime number

// No output for 20

// Checking the condition: 21 <= 20 (false)

// Condition is false, exiting the loop

// Final Output:

2

3

5

7

11

13

17

19

6. W.A.P that uses a while loop to print the numbers from 1 to 10.

// 6 JavaScript program to print numbers from 1 to 10 using a while loop

// Initializing a variable

*var* i = 1;

// Using a while loop to print numbers from 1 to 10

while (i <= 10) {

    console.log(i);

    i++;

}

//Tracking:

// Initializing a variable

*var* i = 1;

// Checking the condition: 1 <= 10 (true)

// Entering the while loop

// Output: 1

// Incrementing i by 1: i = 2

// Checking the condition: 2 <= 10 (true)

// Entering the while loop

// Output: 2

// Incrementing i by 1: i = 3

// Checking the condition: 3 <= 10 (true)

// Entering the while loop

// Output: 3

// Incrementing i by 1: i = 4

// Checking the condition: 4 <= 10 (true)

// Entering the while loop

// Output: 4

// Incrementing i by 1: i = 5

// Checking the condition: 5 <= 10 (true)

// Entering the while loop

// Output: 5

// Incrementing i by 1: i = 6

// Checking the condition: 6 <= 10 (true)

// Entering the while loop

// Output: 6

// Incrementing i by 1: i = 7

// Checking the condition: 7 <= 10 (true)

// Entering the while loop

// Output: 7

// Incrementing i by 1: i = 8

// Checking the condition: 8 <= 10 (true)

// Entering the while loop

// Output: 8

// Incrementing i by 1: i = 9

// Checking the condition: 9 <= 10 (true)

// Entering the while loop

// Output: 9

// Incrementing i by 1: i = 10

// Checking the condition: 10 <= 10 (true)

// Entering the while loop

// Output: 10

// Incrementing i by 1: i = 11

// Checking the condition: 11 <= 10 (false)

// Condition is false, exit the while loop

// The output of the code will be:

1

2

3

4

5

6

7

8

9

10

7. W.A.P that uses a while loop to print the numbers from 10 to 1.

// 7. JavaScript program to print numbers from 10 to 1 using a while loop

// Initializing a variable

*var* i = 10;

// Using a while loop to print numbers from 10 to 1

while (i >= 1) {

    console.log(i);

    i--;

}

//Tracking :

// Initializing a variable

*var* i = 10;

// Checking the condition: 10 >= 1 (true)

// Entering the while loop

// Output: 10

// Decrementing i by 1: i = 9

// Checking the condition: 9 >= 1 (true)

// Entering the while loop

// Output: 9

// Decrementing i by 1: i = 8

// Checking the condition: 8 >= 1 (true)

// Entering the while loop

// Output: 8

// Decrementing i by 1: i = 7

// Checking the condition: 7 >= 1 (true)

// Entering the while loop

// Output: 7

// Decrementing i by 1: i = 6

// Checking the condition: 6 >= 1 (true)

// Entering the while loop

// Output: 6

// Decrementing i by 1: i = 5

// Checking the condition: 5 >= 1 (true)

// Entering the while loop

// Output: 5

// Decrementing i by 1: i = 4

// Checking the condition: 4 >= 1 (true)

// Entering the while loop

// Output: 4

// Decrementing i by 1: i = 3

// Checking the condition: 3 >= 1 (true)

// Entering the while loop

// Output: 3

// Decrementing i by 1: i = 2

// Checking the condition: 2 >= 1 (true)

// Entering the while loop

// Output: 2

// Decrementing i by 1: i = 1

// Checking the condition: 1 >= 1 (true)

// Entering the while loop

// Output: 1

// Decrementing i by 1: i = 0

// Checking the condition: 0 >= 1 (false)

// Condition is false, exit the while loop

// The output of the code will be:

10

9

8

7

6

5

4

3

2

1

8. W.A.P that prints all even numbers between 1 and 20 using a while loop.

// 8. JavaScript program to print even numbers between 1 and 20 using a while loop

// Initializing a variable

*var* i = 2;

// Using a while loop to print even numbers between 1 and 20

while (i <= 20) {

    console.log(i);

    i += 2; // Incrementing by 2 to get the next even number

}

//Tracking:

// Initializing a variable

*var* i = 2;

// Checking the condition: 2 <= 20 (true)

// Entering the while loop

// Output: 2

// Incrementing i by 2: i = 4

// Checking the condition: 4 <= 20 (true)

// Entering the while loop

// Output: 4

// Incrementing i by 2: i = 6

// Checking the condition: 6 <= 20 (true)

// Entering the while loop

// Output: 6

// Incrementing i by 2: i = 8

// Checking the condition: 8 <= 20 (true)

// Entering the while loop

// Output: 8

// Incrementing i by 2: i = 10

// Checking the condition: 10 <= 20 (true)

// Entering the while loop

// Output: 10

// Incrementing i by 2: i = 12

// Checking the condition: 12 <= 20 (true)

// Entering the while loop

// Output: 12

// Incrementing i by 2: i = 14

// Checking the condition: 14 <= 20 (true)

// Entering the while loop

// Output: 14

// Incrementing i by 2: i = 16

// Checking the condition: 16 <= 20 (true)

// Entering the while loop

// Output: 16

// Incrementing i by 2: i = 18

// Checking the condition: 18 <= 20 (true)

// Entering the while loop

// Output: 18

// Incrementing i by 2: i = 20

// Checking the condition: 20 <= 20 (true)

// Entering the while loop

// Output: 20

// Incrementing i by 2: i = 22

// Checking the condition: 22 <= 20 (false)

// Condition is false, exit the while loop

// The output of the code will be:

2

4

6

8

10

12

14

16

18

20

9. W.A.P that prints the sum of numbers from 1 to 10 using a while loop.

// 9. JavaScript program to print the sum of numbers from 1 to 10 using a while loop

// Initializing variables

*var* i = 1;

*var* sum = 0;

// Using a while loop to calculate the sum of numbers from 1 to 10

while (i <= 10) {

    sum += i; // Adding the current value of i to the sum

    i++;      // Incrementing i for the next iteration

}

// Printing the sum to the console

console.log("The sum of numbers from 1 to 10 is: " + sum);

//Tracking :

// Initializing variables

*var* i = 1;

*var* sum = 0;

// Checking the condition: 1 <= 10 (true)

// Entering the while loop

// Adding 1 to the sum: 0 + 1 = 1

// Incrementing i by 1: i = 2

// Checking the condition: 2 <= 10 (true)

// Entering the while loop

// Adding 2 to the sum: 1 + 2 = 3

// Incrementing i by 1: i = 3

// Checking the condition: 3 <= 10 (true)

// Entering the while loop

// Adding 3 to the sum: 3 + 3 = 6

// Incrementing i by 1: i = 4

// Checking the condition: 4 <= 10 (true)

// Entering the while loop

// Adding 4 to the sum: 6 + 4 = 10

// Incrementing i by 1: i = 5

// Checking the condition: 5 <= 10 (true)

// Entering the while loop

// Adding 5 to the sum: 10 + 5 = 15

// Incrementing i by 1: i = 6

// Checking the condition: 6 <= 10 (true)

// Entering the while loop

// Adding 6 to the sum: 15 + 6 = 21

// Incrementing i by 1: i = 7

// Checking the condition: 7 <= 10 (true)

// Entering the while loop

// Adding 7 to the sum: 21 + 7 = 28

// Incrementing i by 1: i = 8

// Checking the condition: 8 <= 10 (true)

// Entering the while loop

// Adding 8 to the sum: 28 + 8 = 36

// Incrementing i by 1: i = 9

// Checking the condition: 9 <= 10 (true)

// Entering the while loop

// Adding 9 to the sum: 36 + 9 = 45

// Incrementing i by 1: i = 10

// Checking the condition: 10 <= 10 (true)

// Entering the while loop

// Adding 10 to the sum: 45 + 10 = 55

// Incrementing i by 1: i = 11

// Checking the condition: 11 <= 10 (false)

// Condition is false, exit the while loop

// The output of the code will be:

The sum of numbers from 1 to 10 is: 55

10. W.A.P that prints the prime numbers 1 to 20 using a while loop.

// 10 JavaScript program to print prime numbers between 1 and 20 using a while loop

// Function to check if a number is prime

*function* isPrime(*num*) {

    if (*num* <= 1) return false;

*var* i = 2;

    while (i <= Math.sqrt(*num*)) {

        if (*num* % i === 0) {

            return false;

        }

        i++;

    }

    return true;

}

// Initializing i to 1

*var* i = 1;

// Using a while loop to iterate from 1 to 20

while (i <= 20) {

    if (isPrime(i)) {

        console.log(i);

    }

    i++;

}

// Initializing i to 1

*var* i = 1;

// Checking the condition: 1 <= 20 (true)

// Entering the while loop

// i = 1

// Checking if 1 is prime (false)

// Incrementing i by 1: i = 2

//No output for 1

// Checking the condition: 2 <= 20 (true)

// Entering the while loop

// Checking if 2 is prime (true)

// Output: 2

// Incrementing i by 1: i = 3

// Checking the condition: 3 <= 20 (true)

// Entering the while loop

// Checking if 3 is prime (true)

// Output: 3

// Incrementing i by 1: i = 4

// Checking the condition:4 <= 20 (true)

// Entering the while loop

// Checking if 4 is prime (false)

// No Output: 4

// Incrementing i by 1: i = 5

// Checking the condition: 5 <= 20 (true)

// Entering the while loop

// Checking if 5 is prime (true)

// Output: 5

// Incrementing i by 1: i = 6

// Checking the condition: 6 <= 20 (true)

// Entering the while loop

// Checking if 6 is prime (false)

// No Output: 6

// Incrementing i by 1: i = 7

// Checking the condition: 7 <= 20 (true)

// Entering the while loop

// Checking if 7 is prime (true)

// Output: 7

// Incrementing i by 1: i = 8

// Checking the condition: 8 <= 20 (true)

// Entering the while loop

// Checking if 8 is prime (false)

// No Output: 8

// Incrementing i by 1: i = 9

// Checking the condition: 9 <= 20 (true)

// Entering the while loop

// Checking if 9 is prime (false)

// No Output: 9

// Incrementing i by 1: i = 10

// Checking the condition: 10 <= 20 (true)

// Entering the while loop

// Checking if 10 is prime (false)

// No Output: 10

// Incrementing i by 1: i = 11

// Checking the condition: 11 <= 20 (true)

// Entering the while loop

// Checking if 11 is prime (true)

// Output: 11

// Incrementing i by 1: i = 12

// Checking the condition: 12 <= 20 (true)

// Entering the while loop

// Checking if 12 is prime (false)

// No Output: 12

// Incrementing i by 1: i = 13

// Checking the condition: 13 <= 20 (true)

// Entering the while loop

// Checking if 13 is prime (true)

// Output: 13

// Incrementing i by 1: i = 14

// Checking the condition: 14 <= 20 (true)

// Entering the while loop

// Checking if 14 is prime (false)

// No Output: 14

// Incrementing i by 1: i = 15

// Checking the condition: 15 <= 20 (true)

// Entering the while loop

// Checking if 15 is prime (false)

// No Output: 15

// Incrementing i by 1: i = 16

// Checking the condition: 16 <= 20 (true)

// Entering the while loop

// Checking if 16 is prime (false)

// No Output: 16

// Incrementing i by 1: i = 17

// Checking the condition: 17 <= 20 (true)

// Entering the while loop

// Checking if 17 is prime (true)

// Output: 17

// Incrementing i by 1: i = 18

// Checking the condition: 18 <= 20 (true)

// Entering the while loop

// Checking if 18 is prime (false)

//  No Output: 18

// Incrementing i by 1: i = 19

// Checking the condition: 19 <= 20 (true)

// Entering the while loop

// Checking if 19 is prime (true)

// Output: 19

// Incrementing i by 1: i = 20

// Checking the condition: 20 <= 20 (true)

// Entering the while loop

// Checking if 20 is prime (false)

//  No Output: 20

// Incrementing i by 1: i = 21

// Checking the condition: 21 <= 20 (false)

// Condition is false, exit the while loop

// The output of the code will be:

2

3

5

7

11

13

17

19

11. W.A.P that uses a do-while loop to print the numbers from 1 to 10.

//Tracking:

// Initializing a variable

*var* i = 1;

// Entering the do-while loop

do {

    // Outputting the current value of i

    console.log(i);

    // Incrementing i by 1 for the next iteration

    i++;

    // Checking the condition: i <= 10 (true)

    // Continuing to the next iteration

} while (i <= 10);

// Checking the condition: i <= 10 (true)

// Entering the do-while loop

// Output: 1

// Incrementing i by 1: i = 2

// Checking the condition: 2 <= 10 (true)

// Entering the do-while loop

// Output: 2

// Incrementing i by 1: i = 3

// Checking the condition: 3 <= 10 (true)

// Entering the do-while loop

// Output: 3

// Incrementing i by 1: i = 4

// Checking the condition: 5 <= 10 (true)

// Entering the do-while loop

// Output: 5

// Incrementing i by 1: i = 5

// Checking the condition: 6 <= 10 (true)

// Entering the do-while loop

// Output: 6

// Incrementing i by 1: i = 6

// Checking the condition: 7 <= 10 (true)

// Entering the do-while loop

// Output: 7

// Incrementing i by 1: i = 7

// Checking the condition: 8 <= 10 (true)

// Entering the do-while loop

// Output: 8

// Incrementing i by 1: i = 8

// Checking the condition: 9 <= 10 (true)

// Entering the do-while loop

// Output: 9

// Incrementing i by 1: i = 9

// Checking the condition: 10 <= 10 (true)

// Entering the do-while loop

// Output: 10

// Incrementing i by 1: i = 10

// Checking the condition: 11 <= 10 (false)

// Condition is false, exit the do-while loop

// The output of the code will be:

1

2

3

4

5

6

7

8

9

10

12. W.A.P that uses a do-while loop to print the numbers from 10 to 1.

// 12 JavaScript program to print numbers from 10 to 1 using a do-while loop

// Initializing a variable

*var* i = 10;

// Using a do-while loop to print numbers from 10 to 1

do {

  console.log(i);

  i--;

} while (i >= 1);

//Tracking:

// Initializing a variable

*var* i = 10;

// Entering the do-while loop

do {

  // Outputting the current value of i

  console.log(i);

  // Decrementing i by 1 for the next iteration

  i--;

  // Checking the condition: i >= 1 (true)

  // Continuing to the next iteration

} while (i >= 1);

// Checking the condition: i >= 1 (true)

// Entering the do-while loop

// Output: 10

// Decrementing i by 1: i = 9

// Checking the condition: 9 >= 1 (true)

// Entering the do-while loop

// Output: 9

// Decrementing i by 1: i = 8

// Checking the condition: 8 >= 1 (true)

// Entering the do-while loop

// Output: 8

// Decrementing i by 1: i = 7

// Checking the condition: 7 >= 1 (true)

// Entering the do-while loop

// Output: 7

// Decrementing i by 1: i = 6

// Checking the condition: 6 >= 1 (true)

// Entering the do-while loop

// Output: 6

// Decrementing i by 1: i = 5

// Checking the condition: 5 >= 1 (true)

// Entering the do-while loop

// Output: 5

// Decrementing i by 1: i = 4

// Checking the condition: 4 >= 1 (true)

// Entering the do-while loop

// Output: 4

// Decrementing i by 1: i = 3

// Checking the condition: 3 >= 1 (true)

// Entering the do-while loop

// Output: 3

// Decrementing i by 1: i = 2

// Checking the condition: 2 >= 1 (true)

// Entering the do-while loop

// Output: 2

// Decrementing i by 1: i = 1

// Checking the condition: 1 >= 1 (true)

// Entering the do-while loop

// Output: 1

// Decrementing i by 1: i = 0

// Checking the condition: 0 >= 1 (false)

// Condition is false, exit the do-while loop

// The output of the code will be:

10

9

8

7

6

5

4

3

2

1

13. W.A.P that prints all even numbers between 1 and 20 using a do while loop.

// 13 JavaScript program to print even numbers between 1 and 20 using a do-while loop

// Initializing a variable

*var* i = 2;

// Using a do-while loop to print even numbers between 1 and 20

do {

  console.log(i);

  i += 2;

} while (i <= 20);

//Tracking:

// Initializing a variable

*var* i = 2;

// Entering the do-while loop

do {

    // Outputting the current value of i (even number)

    console.log(i);

    // Incrementing i by 2 for the next even number

    i += 2;

    // Checking the condition: i <= 20 (true)

    // Continuing to the next iteration

} while (i <= 20);

// Checking the condition: i <= 20 (true)

// Entering the do-while loop

// Output: 2

// Incrementing i by 2: i = 4

// Checking the condition: 4 <= 20 (true)

// Entering the do-while loop

// Output: 4

// Incrementing i by 2: i = 6

// Checking the condition: 6 <= 20 (true)

// Entering the do-while loop

// Output: 6

// Incrementing i by 2: i = 8

// Checking the condition: 8 <= 20 (true)

// Entering the do-while loop

// Output: 8

// Incrementing i by 2: i = 10

// Checking the condition: 10 <= 20 (true)

// Entering the do-while loop

// Output: 10

// Incrementing i by 2: i = 12

// Checking the condition: 12 <= 20 (true)

// Entering the do-while loop

// Output: 12

// Incrementing i by 2: i = 14

// Checking the condition: 14 <= 20 (true)

// Entering the do-while loop

// Output: 14

// Incrementing i by 2: i = 16

// Checking the condition: 16 <= 20 (true)

// Entering the do-while loop

// Output: 16

// Incrementing i by 2: i = 18

// Checking the condition: 18 <= 20 (true)

// Entering the do-while loop

// Output: 18

// Incrementing i by 2: i = 20

// Checking the condition: 20 <= 20 (true)

// Entering the do-while loop

// Output: 20

// Incrementing i by 2: i = 22

// Checking the condition: 22 <= 20 (false)

// Condition is false, exit the do-while loop

// The output of the code will be:

2

4

6

8

10

12

14

16

18

20

14. W.A.P that prints the sum of numbers from 1 to 10 using a do while loop.

// 14 JavaScript program to print the sum of numbers from 1 to 10 using a do-while loop

// Initializing variables

*var* i = 1;

*var* sum = 0;

// Using a do-while loop to calculate the sum of numbers from 1 to 10

do {

    sum += i; // Adding the current value of i to the sum

    i++;     // Incrementing i for the next iteration

} while (i <= 10);

// Printing the sum to the console

console.log("The sum of numbers from 1 to 10 is: " + sum);

//Tracking:

// Initializing variables

*var* i = 1;

*var* sum = 0;

// Entering the do-while loop

do {

    // Adding the current value of i to the sum

    sum += i;

    // Incrementing i by 1 for the next iteration

    i++;

    // Checking the condition: i <= 10 (true)

    // Continuing to the next iteration

} while (i <= 10);

// Checking the condition: i <= 10 (true)

// Entering the do-while loop

// Adding 1 to the sum: 0 + 1 = 1

// Incrementing i by 1: i = 2

// Checking the condition: 2 <= 10 (true)

// Entering the do-while loop

// Adding 2 to the sum: 1 + 2 = 3

// Incrementing i by 1: i = 3

// Checking the condition: 3 <= 10 (true)

// Entering the do-while loop

// Adding 3 to the sum: 3 + 3 = 6

// Incrementing i by 1: i = 4

// Checking the condition: 4 <= 10 (true)

// Entering the do-while loop

// Adding 4 to the sum: 6 + 4 = 10

// Incrementing i by 1: i = 5

// Checking the condition: 5 <= 10 (true)

// Entering the do-while loop

// Adding 5 to the sum: 10 + 5 = 15

// Incrementing i by 1: i = 6

// Checking the condition: 6 <= 10 (true)

// Entering the do-while loop

// Adding 6 to the sum: 15 + 6 = 21

// Incrementing i by 1: i = 7

// Checking the condition: 7 <= 10 (true)

// Entering the do-while loop

// Adding 7 to the sum: 21 + 7 = 28

// Incrementing i by 1: i = 8

// Checking the condition: 8 <= 10 (true)

// Entering the do-while loop

// Adding 8 to the sum: 28 + 8 = 36

// Incrementing i by 1: i = 9

// Checking the condition: 9 <= 10 (true)

// Entering the do-while loop

// Adding 9 to the sum: 36 + 9 = 45

// Incrementing i by 1: i = 10

// Checking the condition: 10 <= 10 (true)

// Entering the do-while loop

// Adding 10 to the sum: 45 + 10 = 55

// Incrementing i by 1: i = 9

// Checking the condition: 11 <= 10 (false)

// Condition is false, exit the do-while loop

// The output of the code will be:

1

2

3

4

5

6

7

8

9

10

The sum of numbers from 1 to 10 is: 55

15. W.A.P that prints the prime numbers 1 to 20 using a do while loop.

// 15 JavaScript program to print prime numbers between 1 and 20 using a do-while loop

// Function to check if a number is prime

*function* isPrime(*num*) {

  if (*num* <= 1) return false;

  for (*var* i = 2; i <= Math.sqrt(*num*); i++) {

    if (*num* % i === 0) {

      return false;

    }

  }

  return true;

}

// Initializing i to 2

*var* i = 2;

// Using a do-while loop to iterate from 2 to 20

do {

  if (isPrime(i)) {

    console.log(i);

  }

  i++;

} while (i <= 20);

//Tracking:

// Initializing i to 2

*var* i = 2;

// Entering the do-while loop

do {

  // Checking if the current value of i is prime using the isPrime function

  if (isPrime(i)) {

    // Outputting the prime number

    console.log(i);

  }

  // Incrementing i by 1 for the next iteration

  i++;

  // Checking the condition: i <= 20 (true)

  // Continuing to the next iteration

} while (i <= 20);

// Checking the condition: 3 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :3

// Incrementing i by 1 for the next iteration i = 4

// Checking the condition: 4 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for : 4

// Incrementing i by 1 for the next iteration i = 5

// Checking the condition: 5 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :5

// Incrementing i by 1 for the next iteration i = 6

// Checking the condition: 6 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :6

// Incrementing i by 1 for the next iteration i = 7

// Checking the condition: 7 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :7

// Incrementing i by 1 for the next iteration i = 8

// Checking the condition: 8 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :8

// Incrementing i by 1 for the next iteration i = 9

// Checking the condition: 9 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :9

// Incrementing i by 1 for the next iteration i = 10

// Checking the condition: 10 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :10

// Incrementing i by 1 for the next iteration i = 11

// Checking the condition: 11 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :11

// Incrementing i by 1 for the next iteration i = 12

// Checking the condition: 12 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :12

// Incrementing i by 1 for the next iteration i = 13

// Checking the condition: 13 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :13

// Incrementing i by 1 for the next iteration i = 14

// Checking the condition: 14 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :14

// Incrementing i by 1 for the next iteration i = 15

// Checking the condition: 15 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :15

// Incrementing i by 1 for the next iteration i = 16

// Checking the condition: 16 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :16

// Incrementing i by 1 for the next iteration i = 17

// Checking the condition: 17 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :17

// Incrementing i by 1 for the next iteration i = 18

// Checking the condition: 18 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :18

// Incrementing i by 1 for the next iteration i = 19

// Checking the condition: 19 <= 20 (true)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// Output :19

// Incrementing i by 1 for the next iteration i = 20

// Checking the condition: 20 <= 20 (false)

// Entering the do-while loop

// Checking if the current value of i is prime using the isPrime function

// Outputting the prime number

// No Output for :20

// Incrementing i by 1 for the next iteration i = 21

// Checking the condition: 21 <= 20 (false)

// Condition is false, exit the do-while loop

// The output of the code will be:

2

3

5

7

11

13

17

19