**Q.1 What is JavaScript. How to use it?**

JavaScript (JS) is a cross-platform, object-oriented programming language used by developers to make web pages interactive.

It allows developers to create dynamically updating content, use animations, pop-up menus, clickable buttons, control multimedia, etc.

**Q.2 How many types of Variables in JavaScript?**

There are two types of variables in JavaScript: local variable and global variable.

There are some rules while declaring a JavaScript variable (also known as identifiers).

Name must start with a letter (a to z or A to Z), underscore (), or dollar ($) sign.

Q.3 Define a Data Types in js?

Data types in JavaScript define the data type that a variable can store.

JavaScript includes primitive and non-primitive data types. The primitive data types in JavaScript include string, number, Boolean, undefined, null, and symbol.

The non-primitive data type includes the object

Q.5 What the deference between undefined and undeclare in JavaScript?

Undefined: It occurs when a variable has been declared but has not been assigned any value. Undefined is not a keyword.

Undeclared: It occurs when we try to access any variable that is not initialized or declared earlier using the var or const keyword.

**Q.6 Using console.log () print out the following statement: The quote 'There is no exercise**

**better for the heart than reaching down and lifting people up.' by John Holmes teaches us to**

**help one another. Using console.log () print out the following quote by Mother Teresa:**

To print out the provided quotes using **console.log ()** in JavaScript, you can use the following code:

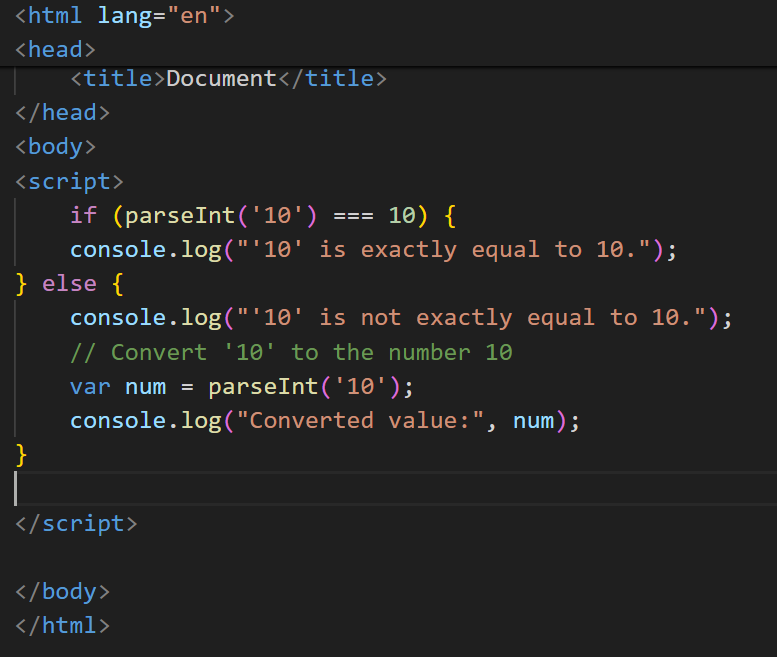
console.log("The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another.");

console.log("The quote 'Not all of us can do great things. But we can do small things with great love.' by Mother Teresa.");

**Q.7 Check if typeof '10' is exactly equal to 10. If not make it exactly equal?**

In JavaScript, the typeof operator returns a string indicating the type of the operand. When you use typeof '10', it will return 'string' because '10' is a string.

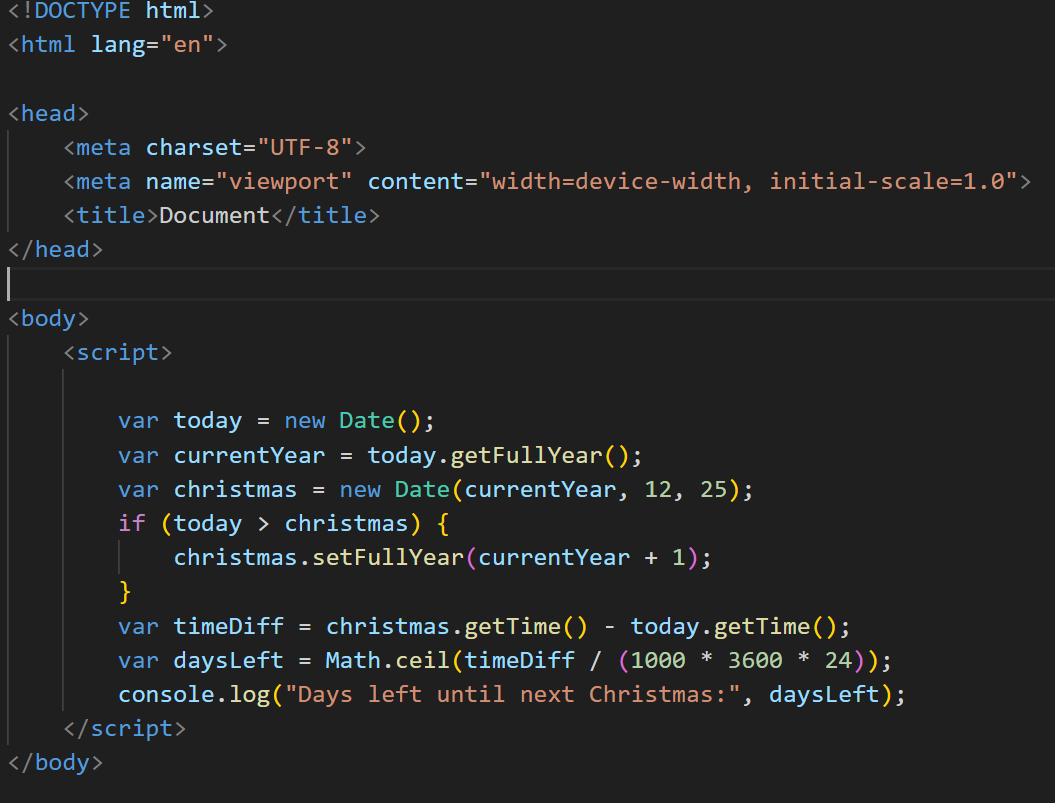
The string '10' can be converted to the number 10, so you use the parseInt() function or the Number() function.

****

**Q.8 Write a JavaScript Program to find the area of a triangle?**

****

**Q.9 Write a JavaScript program to calculate days left until next Christmas?**

****

**Q.10 What is Condition Statement?**

In JavaScript, a condition statement is a programming construct that allows you to execute different blocks of code based on whether a certain condition evaluates to true or false.

1.if statement: The if statement allows you to execute a block of code if a specified condition is true.

if (condition) {

// Code

} else {

// Code

}

2.else if statement: The else if statement allows you to specify multiple conditions to be tested sequentially.

if (condition1) {

// Code

} else if (condition2) {

// Code

} else {

// Code

}

3.switch statement: The switch statement evaluates an expression and executes a block of code associated with a matching case label.

switch (expression) {

case value1:

// Code

break;

case value2:

// Code

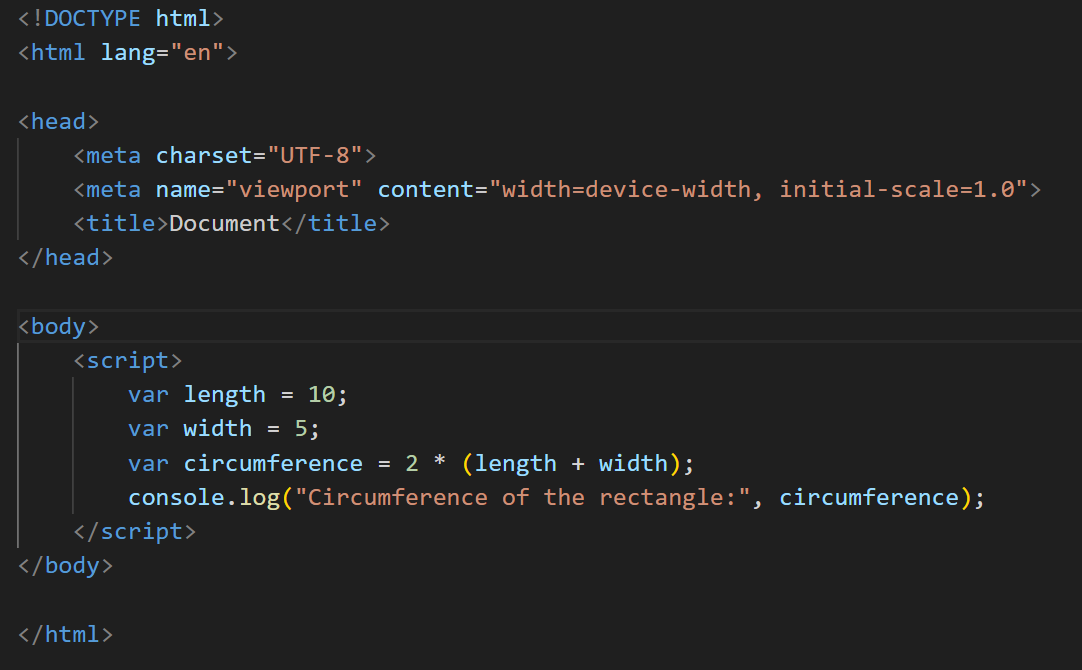
break;

default:

// Code

}

**Q.11 Find circumference of Rectangle** formula **: C = 4 \* a ?**

****

**Q.12 WAP to convert years into days and days into years?**



Q.15 What is the result of the expression (5 > 3 && 2 < 4)?

In JavaScript, the expression (5 > 3 && 2 < 4) involves the logical AND operator (&&).

The logical AND operator returns true if both of its operands are true; otherwise, it returns false.

5 > 3 evaluates to true because 5 is greater than 3.

2 < 4 evaluates to true because 2 is less than 4.

Since both conditions (5 > 3 and 2 < 4) are true, the expression (5 > 3 && 2 < 4) evaluates to true.

So, the result of the expression (5 > 3 && 2 < 4) in JavaScript is true.

**Q.16 What is the result of the expression (true && 1 && "hello")?**

The expression (true && 1 && "hello") evaluates to "hello".

Here's the breakdown:

true is considered truthy in JavaScript.

1 is also considered truthy.

"hello" is a non-empty string, which is also considered truthy.

In a logical AND operation (&&), if all operands are truthy, the result will be the last truthy value encountered. Therefore, the result of (true && 1 && "hello") is "hello".

**Q.17 What is the result of the expression true && false || false && true?**

The expression true && false || false && true is evaluated based on operator precedence.

In JavaScript, the logical AND operator (&&) has higher precedence than the logical OR operator (||). Therefore, the expression is evaluated from left to right, with && operations being evaluated before || operations.

true && false evaluates to false because one of the operands (false) is false.

false && true also evaluates to false for the same reason.

false || false

In a logical OR operation (||), if any operand is true, the result will be true. However, in this case, both operands are false, so the result will be false.

So, the result of the expression true && false || false && true is false.

**Q.18 What is a Loop and Switch Case in JavaScript define that ?**

In JavaScript, a loop is a programming construct that allows you to execute a block of code repeatedly. There are several types of loops in JavaScript, but the most commonly used ones are the for loop, the while loop, and the do-while loop.

for loop: A for loop repeats a block of code a specified number of times.

for (initialization; condition; iteration) {

// code

}

for (let i = 0; i < 5; i++) {

console.log(i);

}

while loop: A while loop repeats a block of code while a specified condition is true.

while (condition) {

// code

}

let i = 0;

while (i < 5) {

console.log(i);

i++;

}

do-while loop: A do-while loop is similar to a while loop, but it ensures that the block of code is executed at least once before checking the condition

do {

// code

} while (condition);

let i = 0;

do {

console.log(i);

i++;

} while (i < 5);

**Switch:** A switch statement is used to perform different actions based on different conditions. It evaluates an expression, then compares the result with various case labels and executes the code associated with the first matching case.

switch (expression) {

case value1:

// code

break;

case value2:

// code

break;

...

default:

// default code

}

Examlpe:

let fruit = 'Apple';

switch (fruit) {

case 'Banana':

console.log('Banana is yellow.');

break;

case 'Apple':

console.log('Apple is red.');

break;

default:

console.log('Unknown fruit.');

}

**Q.19 What is the use of is Nan function?**

The isNaN() function in JavaScript is used to determine whether a value is NaN (Not-a-Number) or not. NaN is a special value that represents the result of an invalid or undefined mathematical operation, such as dividing zero by zero or trying to convert a string that doesn't represent a number into a number.

isNaN(value);

**Q.20 What is the difference between && and || in JavaScript?**

• In JavaScript, && (logical AND) and || (logical OR) are two different logical operators used for combining conditional statements. Here's a brief explanation of each:

Logical AND (&&):

The && operator returns true if both operands are true, and false otherwise.

It can be used to chain multiple conditions, and the entire expression is true only if all conditions are true

Logical OR (||):

The || operator returns true if at least one of the operands is true, and false if both operands are false.

It is commonly used to provide a fallback or alternative value when one condition fails.

**Q.21 What is the use of Void (0)?**

In JavaScript, void(0) is a way to produce an undefined value. It's often used in conjunction with JavaScript event handlers to ensure that the browser doesn't perform any default action when a link is clicked.

For example, in an HTML anchor tag (<a>), if you set the href attribute to javascript:void(0), clicking the link won't cause the browser to navigate to a new page or perform any other default action.

**47 .What is the drawback of declaring methods directly in JavaScript objects?**

In JavaScript, declaring methods directly in objects can have some drawbacks, primarily related to code organization, inheritance, and memory usage. Here are some potential drawbacks:

Code Organization: When methods are declared directly in objects, it can lead to less organized and harder-to-maintain code, especially as the codebase grows. This approach might make it difficult to separate concerns and follow best practices for modular and maintainable code.

Inability to Inherit: Objects created with methods declared directly are typically standalone instances, and they don't benefit from prototype-based inheritance. This can limit code reuse and make it harder to implement common functionality across multiple objects.

**Q What is the use of isNaN function?**

The isNaN () function in JavaScript is used to determine whether a value is NaN (Not-a-Number) or not. NaN is a special value in JavaScript that represents an invalid number result of an arithmetic operation.

Here's how isNaN() works:

If the argument passed to isNaN () is NaN (i.e., not a number), it returns true.

If the argument is a valid number or a value that can be converted to a number, it returns false.

**Q Which company developed JavaScript?**

* JavaScript was developed by Netscape Communications Corporation, particularly by Brendan Eich in 1995.

**Q What are undeclared and undefined variables?**

In JavaScript, "undeclared" and "undefined" are terms that refer to different states of variables:

Undeclared Variables:

Undeclared variables are those that have not been declared using the var, let, or const keywords. When you attempt to use an undeclared variable, JavaScript will throw a ReferenceError.

Example of an undeclared variable:

console.log(x); // ReferenceError: x is not defined

Undefined Variables:

Undefined variables are those that have been declared but have not been assigned a value, or variables that have been explicitly assigned the value undefined.

Example of an undefined variable:

let y;

console.log(y); // Output: undefined

In this example, y is declared but not assigned a value,

It's important to note that undefined is also a special value in JavaScript that indicates that a variable has not been initialized or assigned a value. When a variable is declared but not assigned a value, its default value is undefined. Additionally, a function without a return statement implicitly returns undefined.

**Q 59 What is Bom vs Dom in JS?**

DOM (Document Object Model)

• The Document Object Model (DOM) is a programming interface for HTML and XML documents.

It represents the structure of the document as a hierarchical tree of nodes, where each

node represents a part of the document (such as elements, attributes, and text).

• The DOM provides methods and properties for interacting with and manipulating the

structure and content of web pages. You can use DOM methods to access, create, modify,

and delete elements and attributes in an HTML document dynamically.

• Example DOM operations include selecting elements by their ID, class, or tag name,

changing element styles, adding event listeners, creating new elements, and more.

2. BOM (Browser Object Model):

• The Browser Object Model (BOM) represents the browser itself as an object, providing

access to browser-specific features and functionalities.

• Unlike the DOM, which deals with the document structure, the BOM deals with