1) What is JavaScript. How to use it?

JavaScript (JS) is a versatile programming language that makes web pages dynamic and interactive. It's one of the core technologies alongside HTML and CSS. Here's how to use it:

* **Client-side scripting:** Embed JS code directly within HTML pages using <script> tags. This code executes in the user's web browser, allowing for real-time manipulation of page elements.
* **Server-side scripting:** Node.js, a popular runtime environment, enables JavaScript execution on the server, facilitating data processing and communication with databases.

2) How many type of Variable in JavaScript?

JavaScript has two main types of variables:

1. **Declared variables:** These are explicitly created using the var, let, or const keywords, followed by a variable name:
   * var (function-scoped, mostly avoided in modern JS due to potential scoping issues)
   * let (block-scoped, preferred for variable declaration)
   * const (block-scoped, used for constants that shouldn't be reassigned)

3) Define a Data Types in js?

JavaScript is dynamically typed, meaning variables don't have predefined data types. The type is determined by the value assigned:

* **Number:** Numeric values (e.g., 10, 3.14)
* **String:** Textual data enclosed in quotes (e.g., "hello")
* **Boolean:** True or false values (true, false)
* **Object:** Complex data structures containing key-value pairs (e.g., { name: "John", age: 30 })
* **Array:** Ordered collections of elements (e.g., [1, "apple", true])
* **Undefined:** Represents a variable declared but not yet assigned a value
* **Null:** Denotes an intentional absence of a value
* **Symbol:** A unique and immutable data type (rarely used in basic JS)

4) Write a mul Function Which will Work Properly When invoked With Following Syntax.

function mul(a, b) {

// Type coercion might be necessary if arguments aren't Numbers

return Number(a) \* Number(b);

}

console.log(mul(2, 3)); // Output: 6

console.log(mul("5", 2)); // Output: 10 (coerces strings to numbers)

5) What the deference between undefined and undeclare in JavaScript?

 **Undefined:** A variable is declared but hasn't been assigned a value. You can check for it using typeof variable === "undefined".

 **Undeclared:** A variable is used without being declared at all. This will result in a ReferenceError when the code tries to access it.

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:

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("Mother Teresa said, 'Spread love everywhere you go. Let everyone you meet feel your love. Fill them with love. The love is not diminished by being shared.'");

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

const strNum = '10';

console.log(typeof strNum); // Output: string

const num = Number(strNum);

console.log(typeof num); // Output: number

console.log(num === 10); // Output: true (strict equality check)

8) Write a JavaScript Program to find the area of a triangle?

function triangleArea(base, height) {

return 0.5 \* base \* height;

}

const base = 10;

const height = 5;

const area = triangleArea(base, height);

console.log("Area of the triangle:", area);

9) Write a JavaScript program to calculate days left until next Christmas?

function daysUntilChristmas() {

const today = new Date();

const christmas = new Date(today.getFullYear(), 11, 25); // December 25th

// Check if Christmas has already passed this year

if (today.getMonth() === 11 && today.getDate() > 25) {

christmas.setFullYear(christmas.getFullYear() + 1); // Set to next year's Christmas

}

const oneDay = 1000 \* 60 \* 60 \* 24; // Milliseconds in a day

const timeDifference = christmas.getTime() - today.getTime();

const daysLeft = Math.ceil(timeDifference / oneDay);

return daysLeft;

}

const days = daysUntilChristmas();

console.log(`${days} days left until Christmas!`);

10) What is Condition Statement?

Condition statements (also called conditional statements) are fundamental building blocks in programming. They allow your code to make decisions based on whether a certain condition is true or false. Here's the basic structure:

if (condition) {

// Code to execute if the condition is true

} else {

// Code to execute if the condition is false (optional)

}

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

There seems to be a misunderstanding. The formula C = 4 \* a calculates the perimeter (total distance around the shape) of a square, not a rectangle's circumference (which applies to circles).

For a rectangle's perimeter, you'd use: P = 2 \* (length + width).

12) WAP to convert years into days and days into years?

function convertYearsToDays(years) {

return years \* 365;

}

function convertDaysToYears(days) {

return days / 365;

}

const years = 10;

const daysInYears = convertYearsToDays(years);

console.log(`${years} years is equal to ${daysInYears} days.`);

const days = 3650;

const yearsInDays = convertDaysToYears(days);

console.log(`${days} days is equal to ${yearsInDays.toFixed(2)} years.`); // Round to two decimal places

13) Convert temperature Fahrenheit to Celsius? (Conditional logic Question)

function fahrenheitToCelsius(fahrenheit) {

const celsius = (fahrenheit - 32) \* 5 / 9;

return celsius;

}

const fahrenheit = 77;

const celsius = fahrenheitToCelsius(fahrenheit);

console.log(`${fahrenheit}°F is equal to ${celsius.toFixed(2)}°C.`);

14) Write a JavaScript exercise to get the extension of a filename.?

function getFileExtension(filename) {

const parts = filename.split('.');

return parts.length > 1 ? parts[parts.length - 1] : "";

}

const filename1 = "image.jpg";

const filename2 = "report"; // No extension

const extension1 = getFileExtension(filename1);

const extension2 = getFileExtension(filename2);

console.log(`The extension of ${filename1} is ${extension1}.`);

console.log(`The extension of ${filename2} is ${extension2}.`);

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

The result is true. Both conditions (5 > 3 and 2 < 4) are true, and the && (AND) operator returns true only if both conditions are true.

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

The result is "hello". The && operator evaluates expressions from left to right and stops as soon as it encounters a falsy value (here, 0 or false). Since true is truthy, it proceeds to 1, which is also truthy. Finally, it evaluates "hello" and returns it because both conditions are true.

17) What is the result of the expression true && false || false && true?

The result is true. The && operator has higher precedence than || (OR). So, it evaluates true && false first, which is false. However, because || only needs one truthy value for the entire expression to be true, false || false && true becomes false || true, resulting in true.

18) What is a Loop and Switch Case in JavaScript define that ?

**Loops:**

* Loops are control flow statements that repeatedly execute a block of code until a specific condition is met. They're crucial for iterating over data structures (like arrays) or performing tasks multiple times with varying values.

**Types of Loops:**

1. **for loop:** Executes a code block repeatedly based on a counter variable initialization, condition, and increment/decrement:

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

console.log(i); // Output: 0, 1, 2, 3, 4

}

1. **while loop:** Continues executing a code block as long as a condition remains true:

let count = 0;

while (count < 3) {

console.log(count);

count++; // Increment count

}

// Output: 0, 1, 2

1. **do...while loop:** Executes a code block at least once, then continues as long as a condition is true (similar to while, but guaranteed to run at least once):

let i = 0;

do {

console.log(i);

i++;

} while (i < 2);

// Output: 0, 1

**Switch Case:**

* A switch statement is a control flow structure that allows for executing different code blocks based on the value of an expression. It's a more concise alternative to nested if...else if statements for handling multiple conditions:

const grade = 'A';

switch (grade) {

case 'A':

console.log("Excellent!");

break;

case 'B':

console.log("Good job!");

break;

case 'C':

console.log("You can do better.");

break;

default:

console.log("Invalid grade.");

}

19) What is the use of is Nan function?

The isNaN function in JavaScript stands for "is Not a Number." It's used to determine whether the provided value is an actual number (including positive/negative infinity) or not:

console.log(isNaN(10)); // Output: false (number)

console.log(isNaN("hello")); // Output: true (string)

console.log(isNaN(Infinity));// Output: false (positive infinity)

console.log(isNaN(NaN)); // Output: true (NaN itself)

20) What is the difference between && and || in JavaScript?

**&& (AND):**

* + Evaluates expressions from left to right.
  + Returns true only if **all** expressions are truthy (e.g., non-zero numbers, true, non-empty strings).
  + Stops evaluation as soon as a falsy value (e.g., 0, false, empty string) is encountered, returning false.

Example:

console.log((5 > 3) && (2 < 4)); // Output: true (both conditions true)

console.log((false) && ("hello")); // Output: false (first condition false)

**|| (OR):**

* + Evaluates expressions from left to right.
  + Returns true if **at least one** expression is truthy.
  + Stops evaluation as soon as a truthy value is encountered, returning true. Continues if all are falsy, returning false.

Example:

console.log((false) || (true)); // Output: true (second condition true)

console.log((0) || ("hello")); // Output: "hello" (first condition falsy, second truthy)

console.log((false) || (false)); // Output: false (both conditions false)

Remember the order of precedence, where && has higher priority than ||. Use parentheses to control evaluation order if needed.