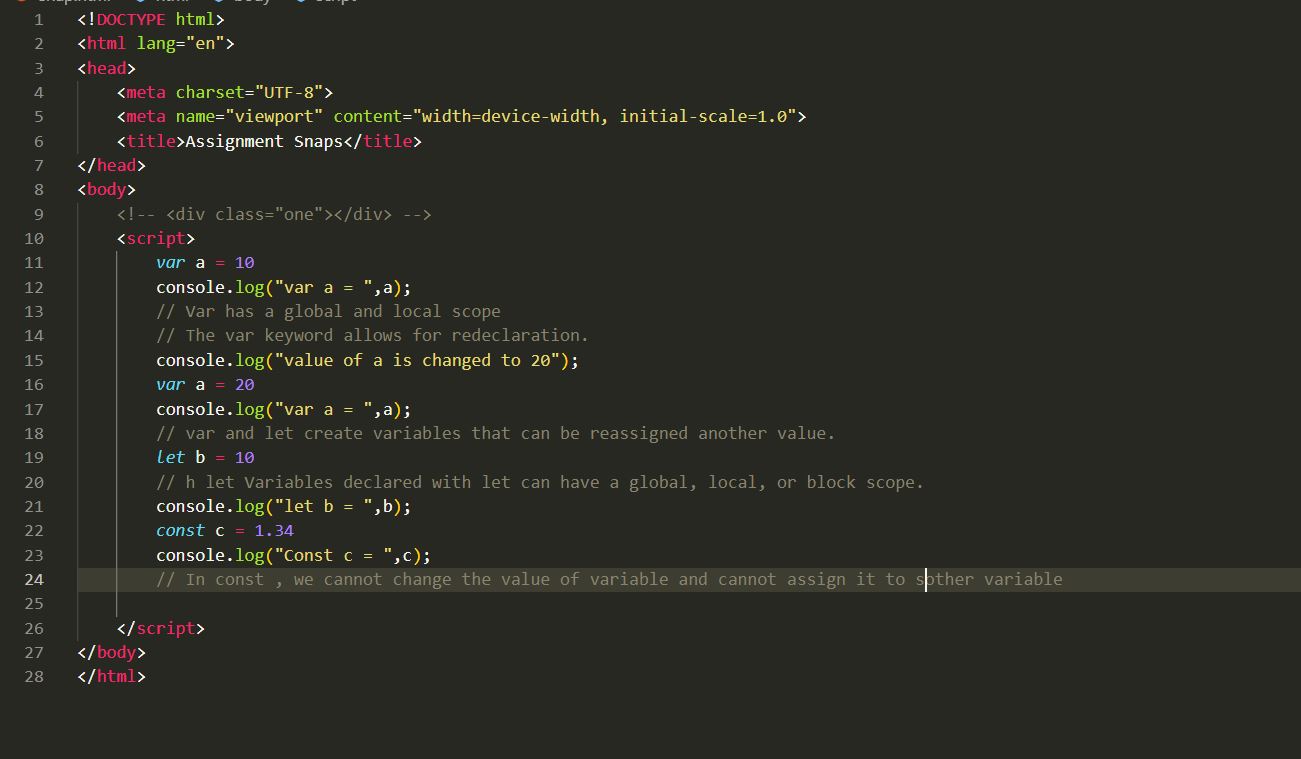
Q1) What is JavaScript. How to use it?

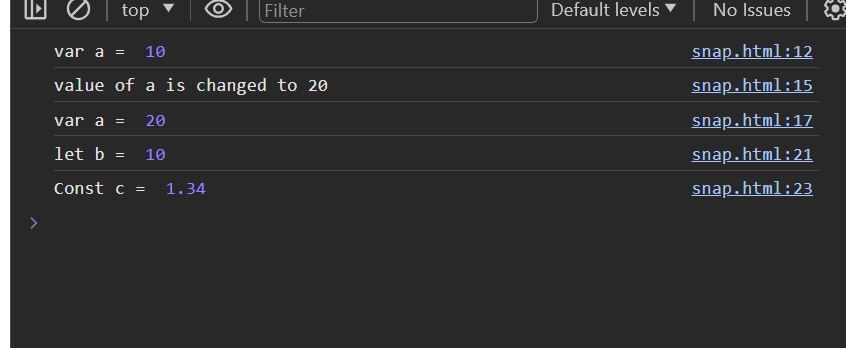
* JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.
* [JavaScript](https://developer.mozilla.org/en-US/docs/Glossary/JavaScript) is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

Q2) How many type of Variable in JavaScript?

* Variables in JavaScript can be declared in 4 ways :

1. Automatically
2. var
3. let
4. const

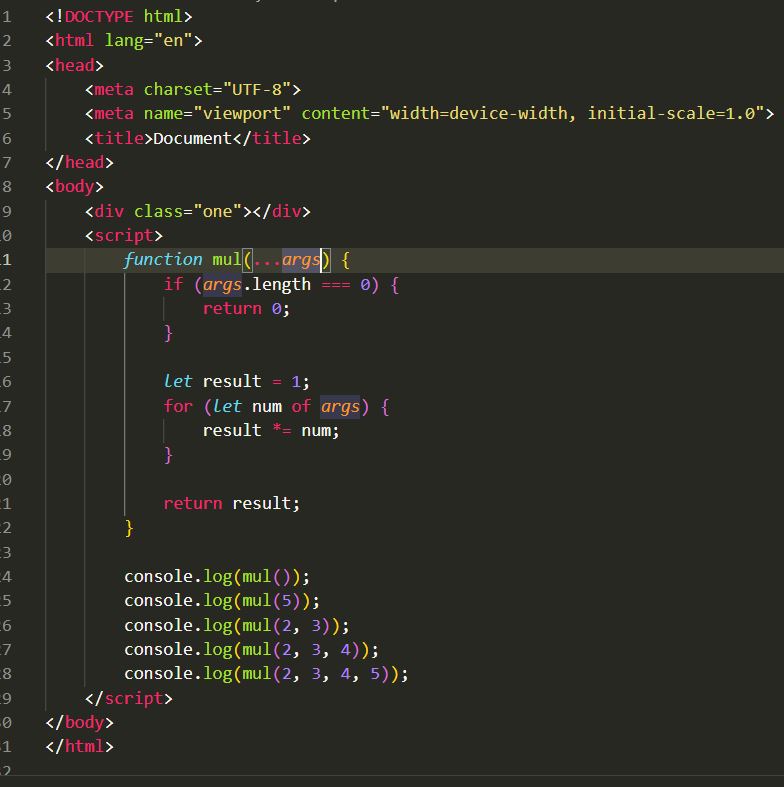


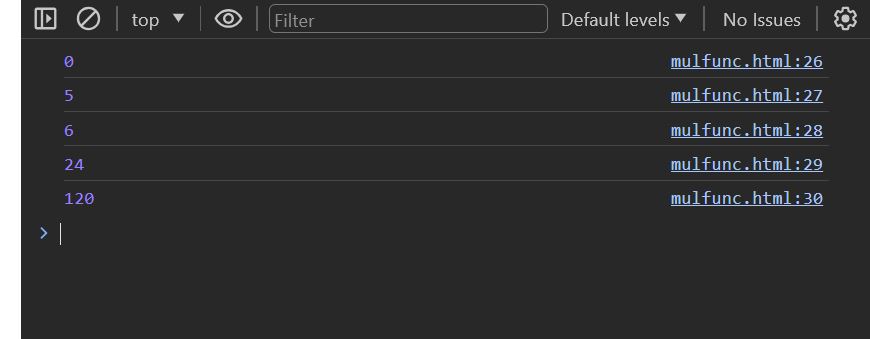


Q3) Define a Data Types in js?

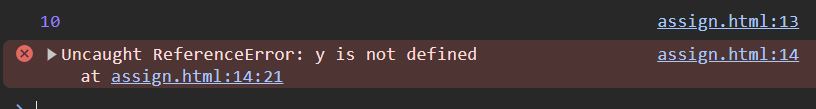
* JavaScript supports several data types including numbers, strings, booleans, arrays, objects, null, and undefined.
* Numbers represent numeric values,
* Strings represent text.
* Booleans represent true or false values.
* Arrays store multiple values in an ordered list.
* Objects store key-value pairs.
* Null represents the intentional absence of any object value, and undefined represents the absence of a defined value.

Q4) Write a mul() Function Which will Work Properly When invoked With Following Syntax.

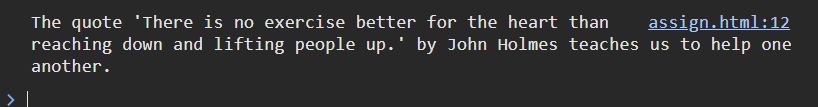
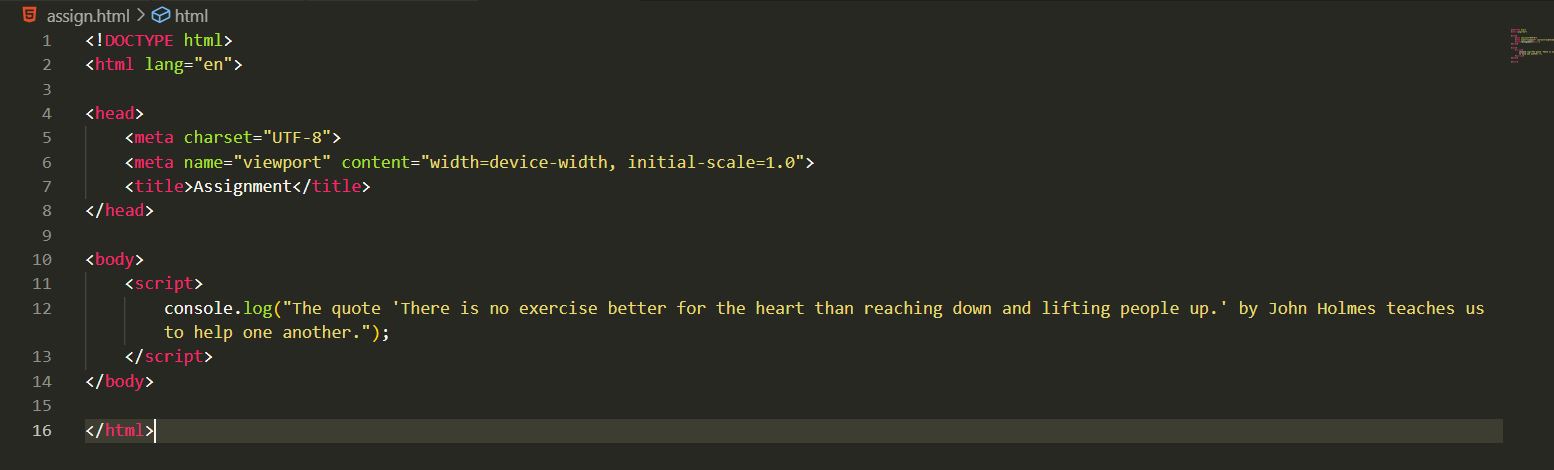




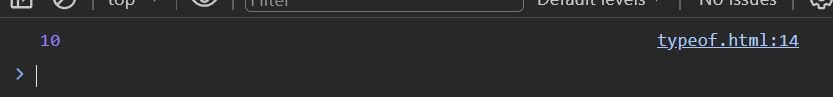
Q5) What the difference between undefined and undeclared in JavaScript?

* 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. If we use ‘typeof’ operator to get the value of an undeclared variable, we will face the runtime error with the return value as “undefined”. The scope of the undeclared variables is always global.
* 

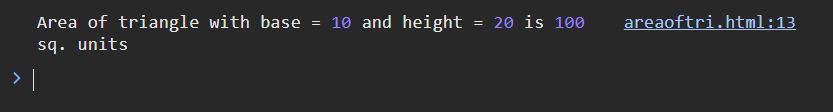
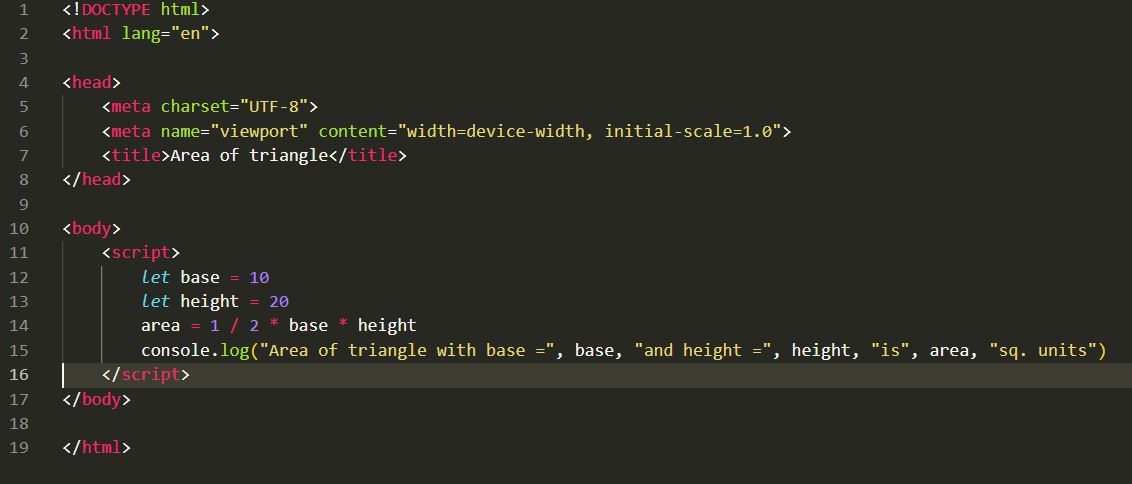
Q6) Using console.log() print out the following statement: The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another. Using console.log() print out the following quote by Mother Teresa:

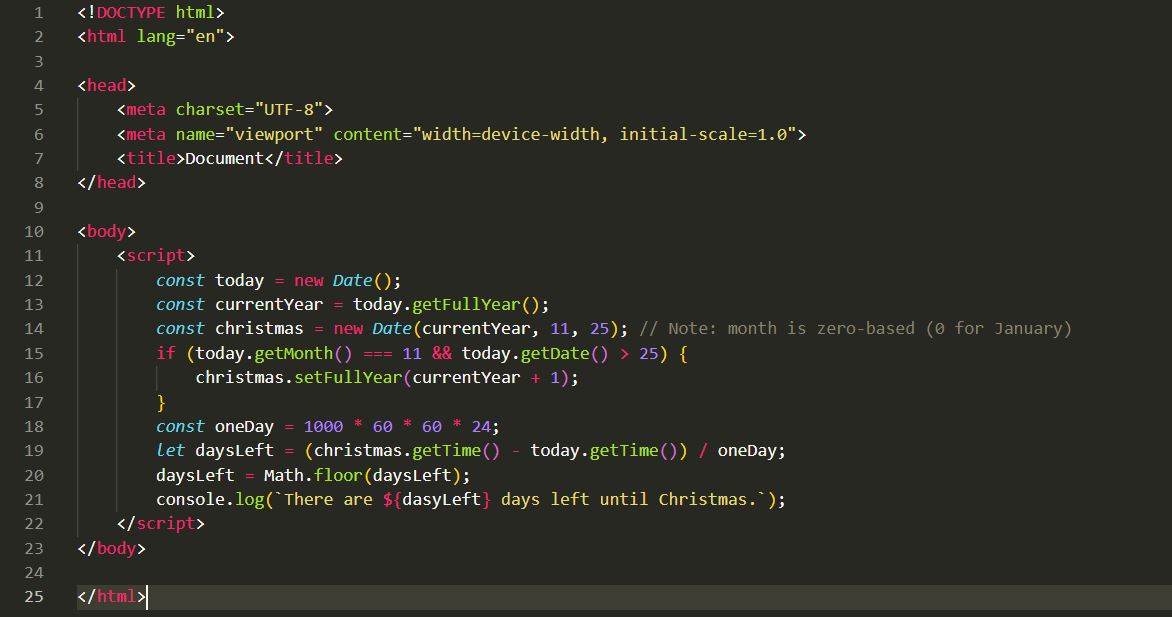
* 

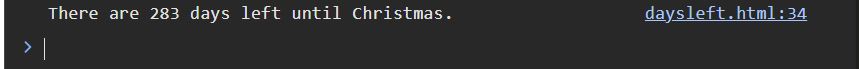
Q7) Check if typeof '10' is exactly equal to 10. If not make it exactly equal?

* 

Q8) Write a JavaScript Program to find the area of a triangle?

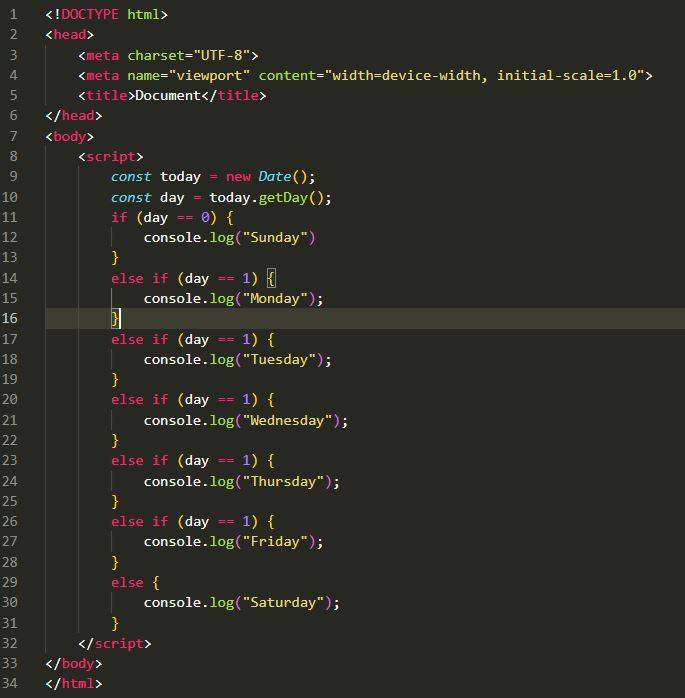
* 

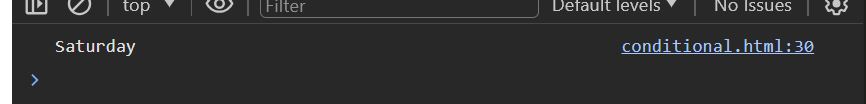
Q9) Write a JavaScript program to calculate days left until next Christmas



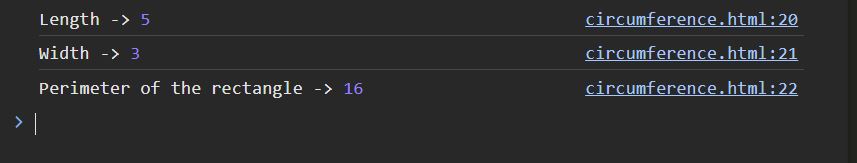
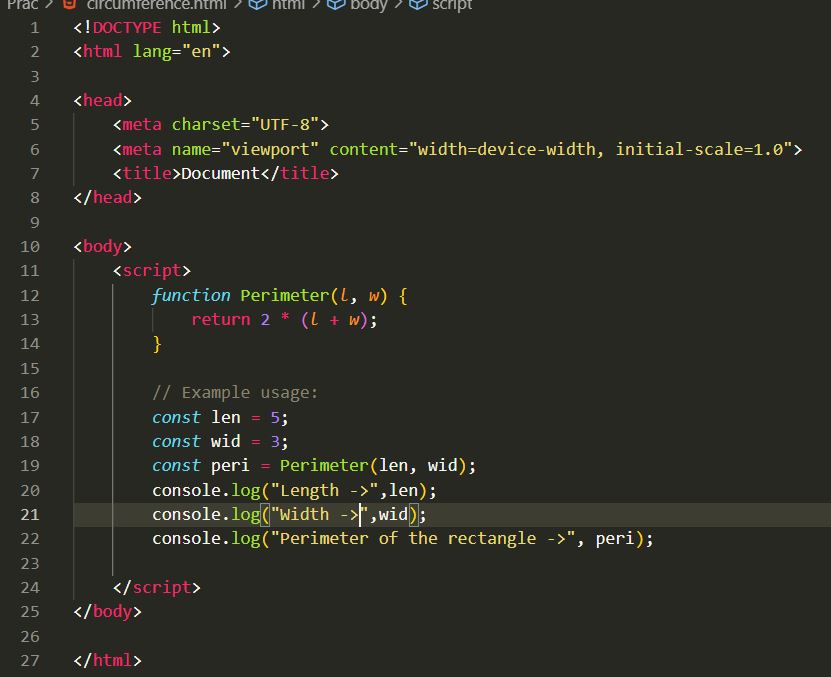
Q10) What is Condition Statement?

* Conditional statements are a powerful tool that can be used to control the flow of your JavaScript code. By using conditional statements, you can make your code more efficient and easier to read.

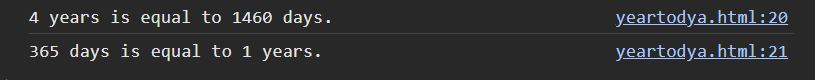
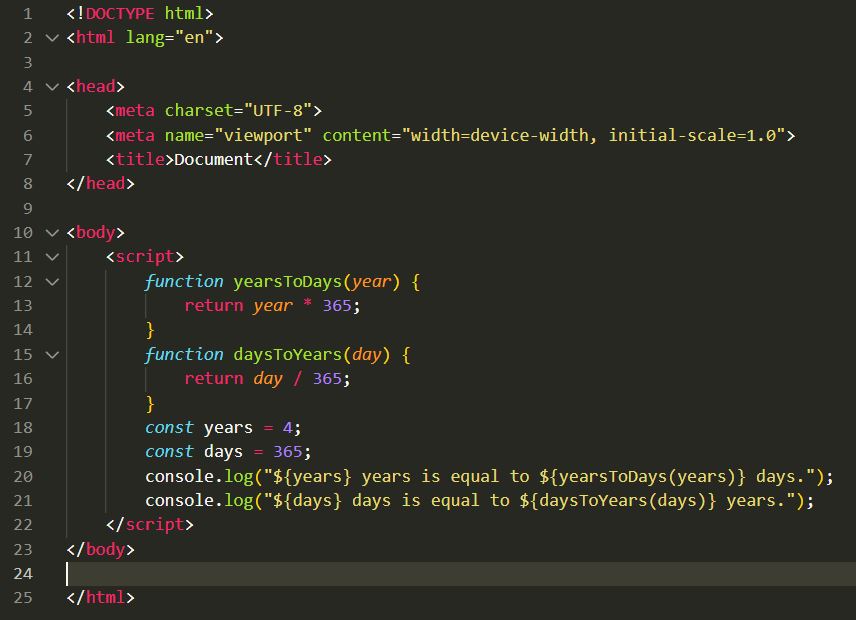




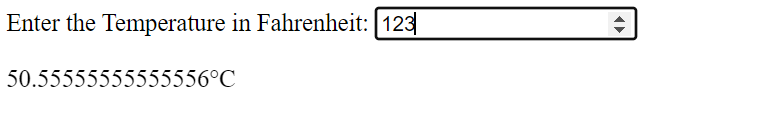
Q11) Find circumference of Rectangle formula : C = 4 \* a ?



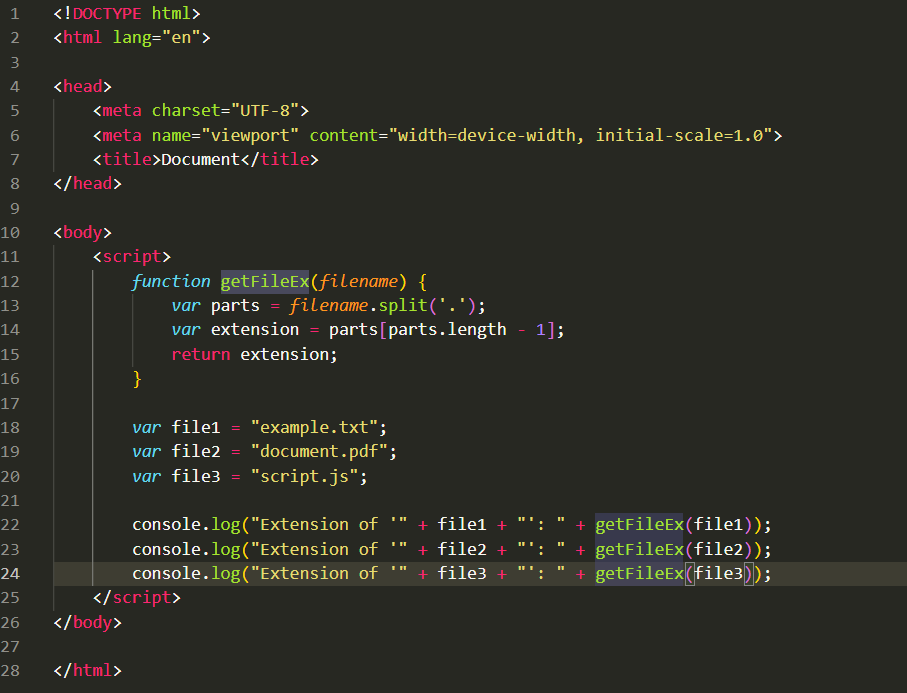
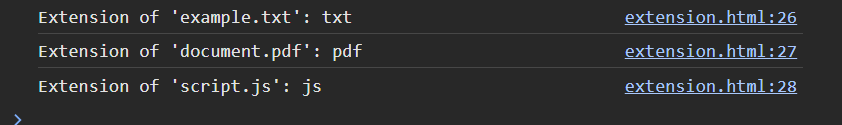
Q12) WAP to convert years into days and days into years?

* 

Q13) Convert temperature Fahrenheit to Celsius? (Conditional logic Question)



Q14) Write a JavaScript exercise to get the extension of a filename.?



Q15) What is the result of the expression (5 > 3 && 2 < 4)?

* True

Q16) What is the result of the expression (true && 1 && "hello")?

* Hello

Q17) What is the result of the expression true && false || false && true?

* false

Q18) What is a Loop and Switch Case in JavaScript define that?

* Loops :

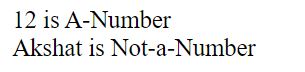
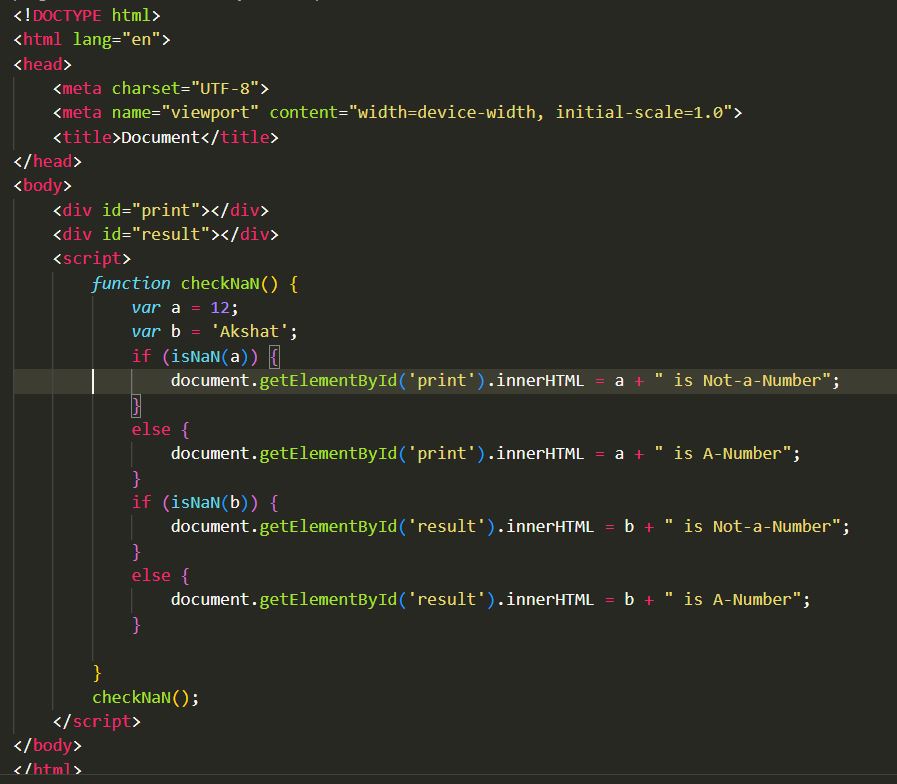
A loop is a programming construct that allows you to repeatedly execute a block of code as long as a specified condition is true. It helps in automating repetitive tasks and iterating over collections of data. JavaScript supports several types of loops, including **for**, **while**, and **do-while** loops.

* Switch Case :

Switch case is a control statement in JavaScript that allows you to execute different blocks of code based on the evaluation of an expression. It provides a cleaner and more efficient alternative to multiple **if-else** statements when dealing with multiple possible conditions. The **switch** statement evaluates an expression and compares it with multiple cases. If a match is found, the corresponding block of code is executed. It also allows a default case to be executed when no other cases match the expression.

Q19) What is the use of is Nan function?

* NaN means ‘Not-a-Number’ in Javascript. It is a function which return true if it is Not a number and false if it is a number.



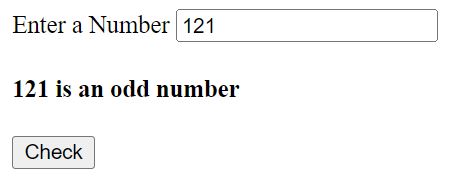
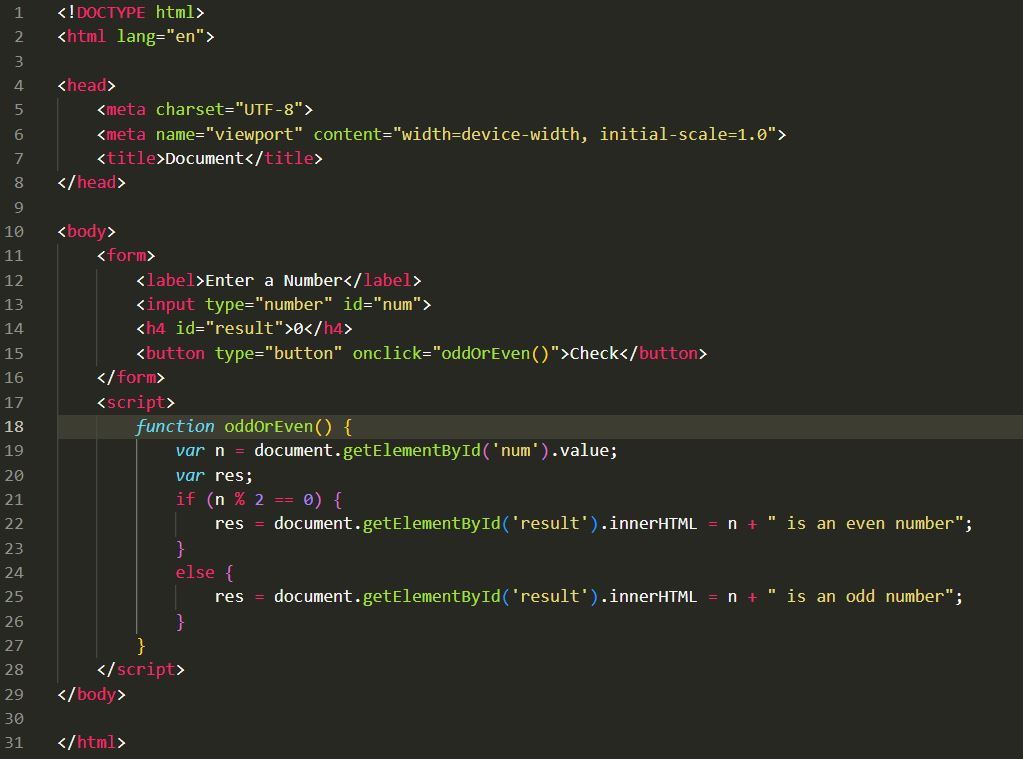
Q20) What is the difference between && and || in JavaScript?

* && and || is used in JavaScript with multiple expressions.
* && stands for ‘and’. It is used to check if all the expressions before and after && are true or not. If even a single expression among multiple expression is false. It will print false.
* || stands for ‘or’. It is used to check if even a single expression is true among all the expressions and if even a single expression is found true it will print true.

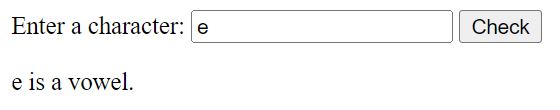
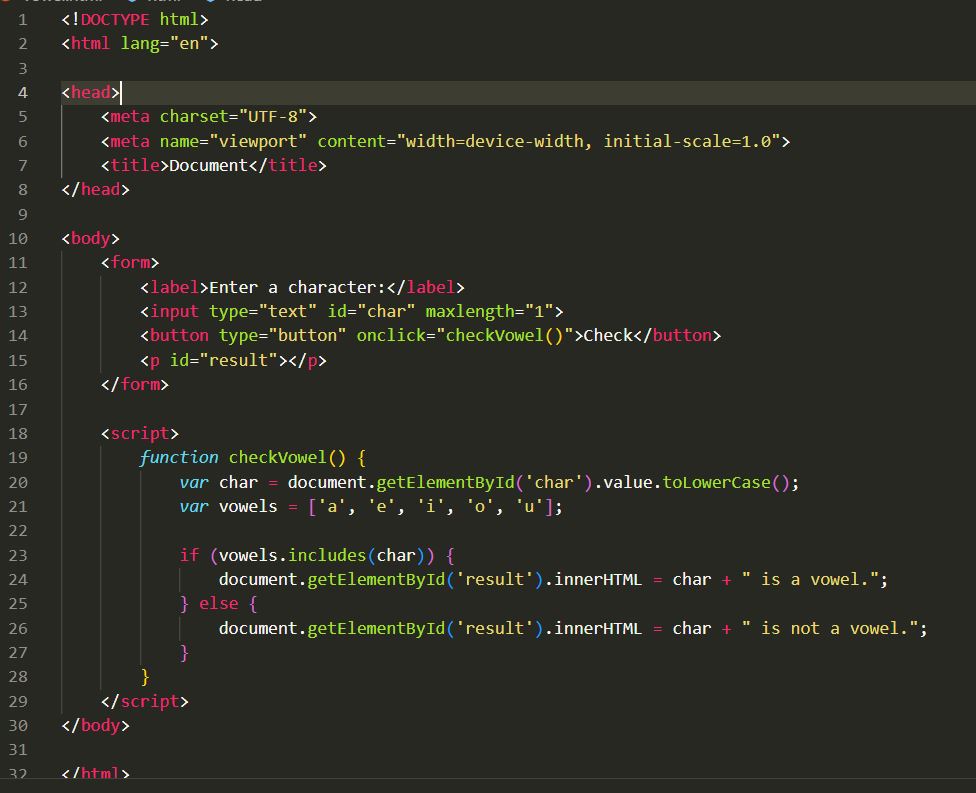
Q21) What is the use of Void (0)?

* The void(0) expression is used in JavaScript primarily to prevent the browser from performing the default action when clicking on a link. It's often used in combination with the href attribute of anchor (<a>) tags to create "dummy" or "void" links that don't navigate anywhere but still trigger some JavaScript functionality.

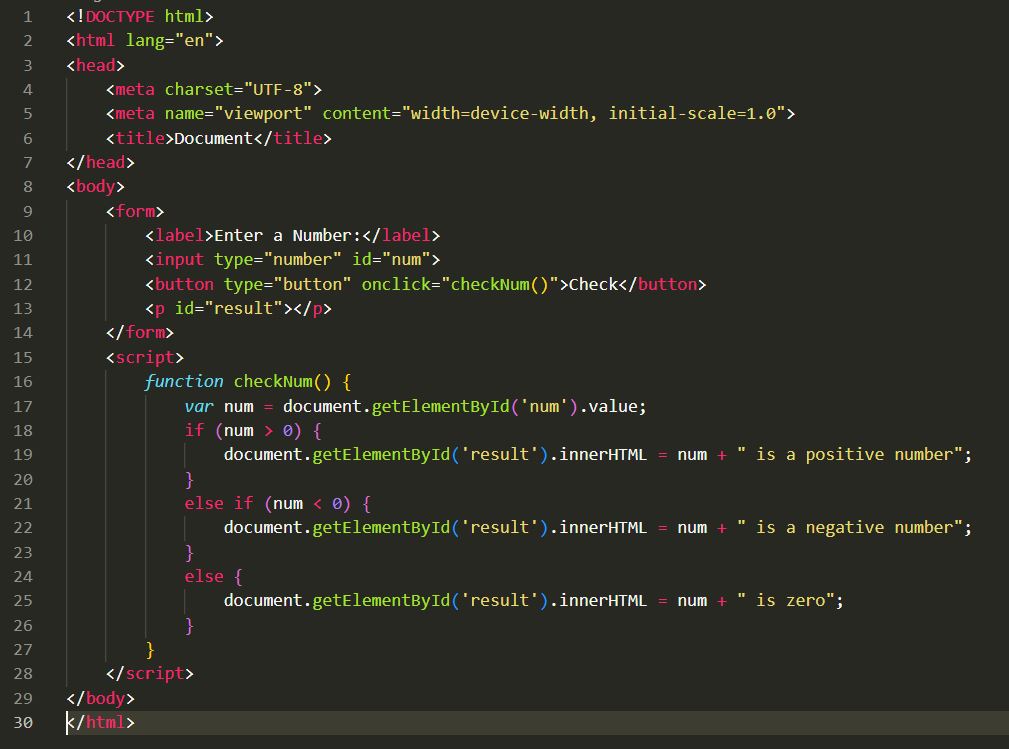
Q22) Check Number Is Positive or Negative in JavaScript?

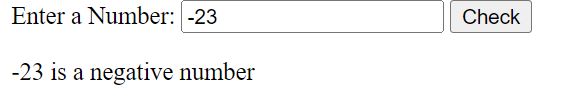


Q23) Find the Character Is Vowel or Not ?



Q24) Write to check whether a number is negative, positive or zero?





Q47) What is the drawback of declaring methods directly in JavaScript objects?

* There are a couple of drawbacks to declaring methods directly in JavaScript objects:
* **Inefficiency:** When you define a method directly within an object, that method gets recreated every time you create a new instance of the object. This can be inefficient, especially for methods that are used frequently.
* **Code Readability and Maintainability:** Having all the methods defined directly within the object can make the code cluttered and harder to understand. It can also be difficult to maintain, as changes to a method would require modifying every single object instance.

Q59) What is BOM vs DOM in JS?

|  |  |  |
| --- | --- | --- |
| **Aspect** | **BOM (Browser Object Model)** | **DOM (Document Object Model)** |
| Purpose | Interacts with the browser's environment | Interacts with the document structure |
| Scope | Browser-specific, not standardized | Standardized across different browsers |
| |  | | --- | | Main Components |  |  | | --- | |  | | window, navigator, screen, history, location | document, elements, attributes, text nodes |
| Functionality | Manages browser windows, user information, navigation | Manipulates HTML/XML content |
| |  | | --- | | Example Use Case |  |  | | --- | |  | | Displaying an alert box, getting browser information | Adding a new element to the HTML document |

Q60) Array vs object defences in JS?

|  |  |  |
| --- | --- | --- |
| Aspect | Array | Objects |
| Definition | An ordered collection of values, indexed by numerical keys (indices). | An unordered collection of key-value pairs, where keys are strings (or symbols) and values can be of any type. |
| |  | | --- | | Use Case |  |  | | --- | |  | | Suitable for storing lists of items that need to be accessed by their order. | Suitable for storing data that needs to be accessed by named keys. |
| Syntax | Declared using square brackets [ ]. | Declared using curly braces { }. |
| |  | | --- | | Accessing Elements |  |  | | --- | |  | | |  | | --- | | Accessed by numerical indices, e.g., array[0]. |  |  | | --- | |  | | |  | | --- | | Accessed by keys, e.g., object.key or object["key"]. |  |  | | --- | |  | |
| Iteration | |  | | --- | | Iterated using loops like for, for...of, forEach(), map(), etc. |  |  | | --- | |  | | |  | | --- | | Iterated using for...in, Object.keys(), Object.values(), Object.entries(), etc. |  |  | | --- | |  | |
| |  | | --- | | Length Property |  |  | | --- | |  | | |  | | --- | | Has a length property that indicates the number of elements. |  |  | | --- | |  | | |  | | --- | | Does not have a length property, but can be approximated using Object.keys().length. |  |  | | --- | |  | |
| Methods | |  | | --- | | Comes with numerous built-in methods like push(), pop(), shift(), unshift(), splice(), slice(), map(), filter(), etc. |  |  | | --- | |  | | |  | | --- | | Methods need to be defined by the user or can use Object methods like Object.keys(), Object.values(), Object.entries(), etc. |  |  | | --- | |  | |
| Performance | |  | | --- | | Optimized for numerical indexing and operations. |  |  | | --- | |  | | |  | | --- | | Generally optimized for quick lookups by keys. |  |  | | --- | |  | |
| Mutability | |  | | --- | | Elements and length can be changed; arrays are mutable. |  |  | | --- | |  | | |  | | --- | | Properties can be added, modified, or removed; objects are mutable. |  |  | | --- | |  | |
| |  | | --- | | Example |  |  | | --- | |  | | |  | | --- | | let arr = [1, 2, 3]; |  |  | | --- | |  | | let obj = {key1: 'value1', key2: 'value2' }; |

Q67) What is JavaScript?

* JavaScript is a **programming language used for creating dynamic content on websites**. It is a**lightweight, cross-platform** and **single-threaded**programming language. JavaScript is an **interpreted**language that executes code line by line providing more flexibility. It is a commonly used programming language to**create dynamic and interactive elements in web applications**. It is easy to learn.

Q68) What is negative Infinity?

* Negative Infinity, denoted as `-Infinity`, is a special value in mathematics and programming that represents a value that is smaller than any other real number. It is the opposite of positive infinity (`+Infinity`). In programming contexts, such as JavaScript, it is often used to represent values that are the smallest possible numeric values or to indicate that an operation results in a value that is too small to be represented or meaningful within the defined numeric range.

Q69) Which company developed JavaScript?

* JavaScript was developed by Netscape Communications Corporation. The language was created by Brendan Eich in 1995 while he was working at Netscape. Initially, it was called Mocha, then renamed to LiveScript, and finally, it was branded as JavaScript to leverage the popularity of Java, even though the two languages are quite different.

Q70) What are undeclared and undefined variables?

* In JavaScript, the terms "undeclared" and "undefined" refer to variables with distinct states. An undeclared variable is one that has not been declared using `var`, `let`, or `const`. Attempting to reference an undeclared variable typically results in a `ReferenceError` because the variable does not exist in the environment. For example, `console.log(x); ` without any prior declaration of `x` will throw an error indicating that `x` is not defined.
* On the other hand, an undefined variable is one that has been declared but has not been assigned a value. Such a variable exists in the scope but has no defined value, and referencing it simply returns `undefined` without throwing an error. For instance, if you declare a variable with `let x; ` and then log its value with `console.log(x);`, it will output `undefined`.
* In summary, undeclared variables do not exist in the execution context and cause errors when accessed, while undefined variables exist but hold no value until they are explicitly assigned one. Understanding this distinction is crucial for debugging and writing effective JavaScript code.

Q71) What is the difference between ViewState and SessionState?

* ViewState and SessionState are both mechanisms for state management in web applications, particularly in ASP.NET. They serve different purposes and have distinct characteristics. Here's a comparison of the two:

|  |  |  |
| --- | --- | --- |
| **Aspect** | **ViewState** | **SessionState** |
| Purpose | Maintains the state of a web page across postbacks | Maintains the state of a user's session across multiple pages |
| Scope | Page-level | Application-level |
| Storage Location | Encoded and stored in a hidden field on the page | Stored on the server, with a session ID sent to the client |
| Data Persistence | Data is persisted only for the duration of the page's life cycle | Data is persisted for the duration of the user session |
| Data Size | Limited to the size of the page | Can store larger amounts of data compared to ViewState |
| |  | | --- | | Security |  |  | | --- | |  | | Data is stored in the page, so it can be tampered with if not encrypted | Data is stored on the server, making it more secure from client-side tampering |
| Usage Example | Storing control values, user input, or page-specific information that needs to persist across postbacks | Storing user-specific information like authentication details, shopping cart contents, or user preferences across multiple pages |

Q72) What are the disadvantages of using innerHTML in JavaScript?

* While innerHTML can be handy for simple tasks, it comes with some drawbacks:
* **Security Risk:** innerHTML allows you to insert any HTML code, including malicious scripts. This can be exploited for Cross-Site Scripting (XSS) attacks, where attackers inject scripts into your page to steal data or redirect users.
* **Performance Issues:** innerHTML can be slow, especially when dealing with large amounts of content. This is because it replaces the entire content of the element, including any existing event listeners, and then re-parses the new HTML.
* **Limited Functionality:** innerHTML isn't ideal for all situations. It can't easily append content or handle complex DOM manipulations.
* **Harder to Maintain:** Code using innerHTML can become messy and difficult to maintain, especially when dealing with multiple elements or complex content.

Q73) How to read and write a file using JavaScript?

* In web browsers, JavaScript primarily operates within a sandboxed environment that restricts direct access to the local file system for security reasons. However, JavaScript can interact with files indirectly through mechanisms like file input elements and APIs that allow interaction with files selected by users.
* **Reading a File (Using File Input Element)**
* To read a file that a user has selected, you can use an <input type="file"> element in your HTML and then handle the file in JavaScript

### Writing to a File

* Directly writing to files from JavaScript in a browser-based environment is not possible due to security restrictions. However, if you are working in a Node.js environment or have access to server-side JavaScript (e.g., using frameworks like Express), you can write files using Node.js file system (fs) module