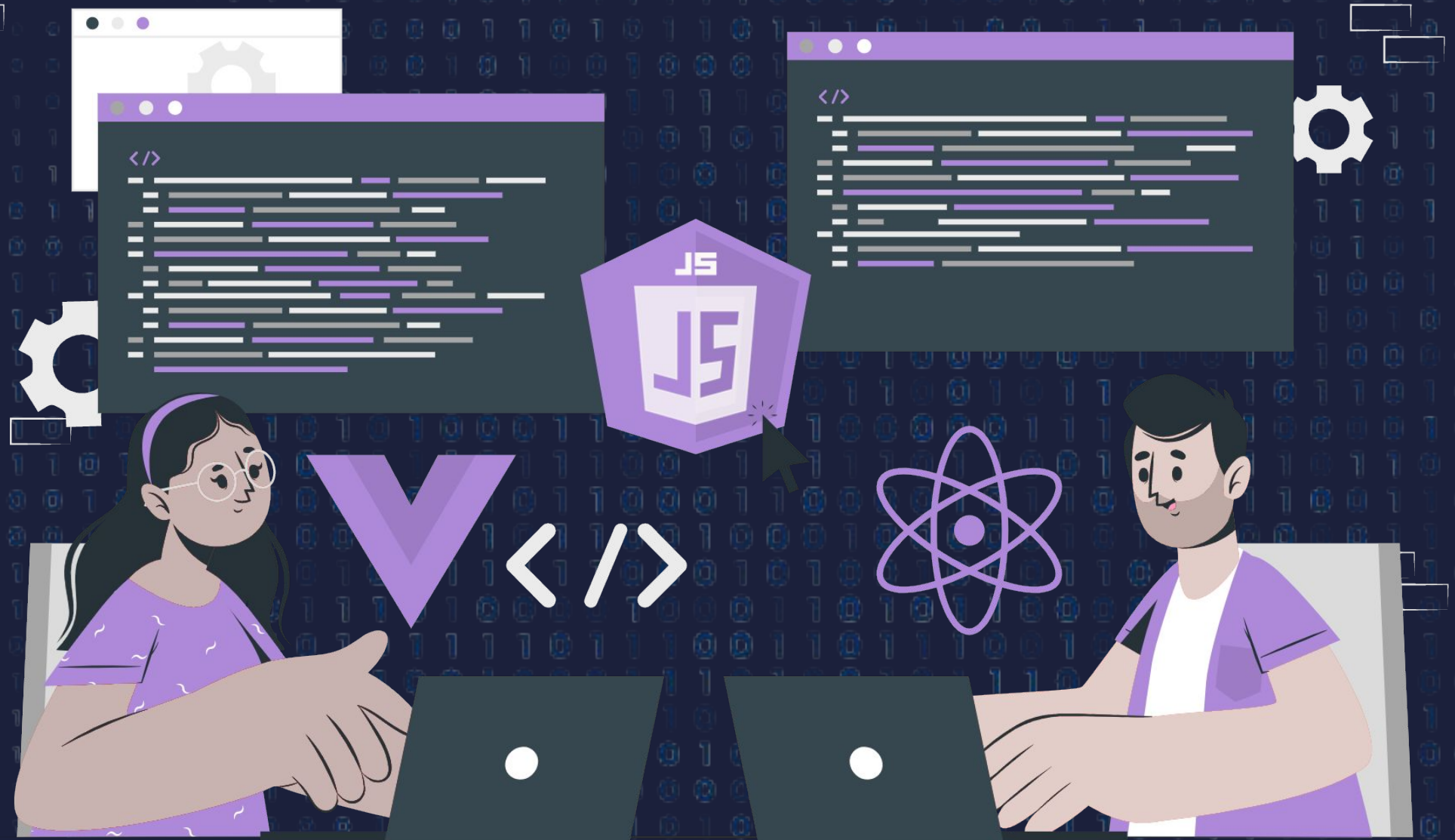




Setting Time (Set Interval and Set timeout)



List of content:

1. setTime()
2. setInterval()
3. clearInterval()
4. setTimeout()
5. clearTimeout()

setTime()

- By adding specified milliseconds to the date January 1, 1970, the JavaScript Date setTime() Function can be used to obtain a date object.
- It takes milliseconds as the parameter and returns an output date by adding these milliseconds to January 1, 1970
- The setTime() function helps to assign a date and time to another Date object.

Example

Set Interval:

- This method repeats a given function at every given time interval.

- **Syntax:**

`window.setInterval(function, milliseconds);`

function: the first parameter is the function to be executed

milliseconds: indicates the length of the time interval between each execution.

Example 1:

Displaying messages with the help of setInterval()

Example 2:

Displaying time with the help of setInterval and another helping function.

Example 3:

Displaying time with the help of setInterval and another helping function.

clearInterval()

- The clearInterval() method cancels a timed, repeating action which was previously established by a call to setInterval().
- If the parameter does not identify a previously established action, this method does nothing.
- **Syntax :**
clearInterval(intervalID)
- **intervalID**
The identifier of the repeated action you want to cancel. This ID was returned by the corresponding call to setInterval().

Example

setTimeout

- The `setTimeout()` method sets a timer that executes a function or specified piece of code once the timer expires.
- **Syntax:**
`setTimeout(function, delay, argument1, argument2.....argumentN)`
- ***function***: is executed after timer expires
- **delay**: in milliseconds is the time for which the timer should wait before the specified function or code is executed. If this parameter is omitted, a value of 0 is used, meaning execute immediately.
- ***arguments***: Additional arguments which are passed through to the function specified here.
- The returned `timeoutID` is a positive integer value which identifies the timer created by the call to `setTimeout()`.
- `setTimeout()` is an ***asynchronous function*** which simply means that the timer function will not pause execution of other functions in the functions stack. More precisely, you cannot use `setTimeout()` to create a "pause" before the next function in the function stack fires.

Example :

Display a set of messages using setTimeout()

clearTimeout()

- The clearTimeout() method cancels a timeout previously established by calling setTimeout().
- **Syntax**
- clearTimeout(timeoutID)
- timeoutID is the identifier of the timeout you want to cancel. This ID was returned by the corresponding call to setTimeout().
- The clearTimeout() method requires the id returned by setTimeout() to know which setTimeout() method to cancel.

Example :

To demonstrate the functioning of `clearTimeout()`



▶ THANK YOU ◀