Shared function

SetTimer

> **Important Note:** The speed at which a client side timer runs can be completely unreliable if a client is maliciously modifying their operating system speed, timers could run much faster or slower.

> **Important Note:** Writing the following code can cause performance issues. Use onClientPreRender instead.
>
> ```
> setTimer(theFunction, 0, 0)
> ```

This function allows you to trigger a function after a number of milliseconds have elapsed. You can call one of your own functions or a built-in function. For example, you could set a timer to spawn a player after a number of seconds have elapsed.

Once a timer has finished repeating, it no longer exists.

The minimum accepted interval is 0ms.

Nexus RP guarantees that the timer will be triggered after *at least* the interval you specify. The resolution of the timer is tied to the frame rate (server side and client-side). All the overdue timers are triggered at a single point each frame. This means that if, for example, the player is running at 30 frames per second, then two timers specified to occur after 100ms and 110ms would more than likely occur during the same frame, as the difference in time between the two timers (10ms) is less than half the length of the frame (33ms). As with most timers provided by other languages, you shouldn't rely on the timer triggering at an exact point in the future.

Syntax

```
timer setTimer ( function theFunction, int timeInterval, int timesToExecute [, var arguments... ] )
```

OOP Syntax Help! I don't understand this!

Method: *Timer(...)*

Required Arguments

- **theFunction:** The function you wish the timer to call.

> **Note:** The hidden global variable **sourceTimer** contains the currently executing timer userdata

> **Note:** The hidden global variable **source** becomes *nil* inside a timer function. You need to declare it on the function arguments if you need to use it.

- **timeInterval:** The number of milliseconds that should elapse before the function is called. The minimum is 0 ms; 1000 milliseconds = 1 second)
- **timesToExecute:** The number of times you want the timer to execute, or 0 for infinite repetitions.

Optional Arguments

NOTE: When using optional arguments, you might need to supply all arguments before the one you wish to use. For more information on optional arguments, see optional arguments.

- **arguments:** Any arguments you wish to pass to the function can be listed after the *timesToExecute* argument. Note that any tables you want to pass will get cloned, whereas metatables and functions/function references in that passed table will get lost. Also changes you make in the original table before the function gets called won't get transferred.

Returns

Returns a timer pointer if the timer was set successfully, *false* if the arguments are invalid or the timer could not be set.