

## Unit 5.2 Highlights

**setTimeout()** - Delayed before a *single* execution of code  
First parameter is a function (callback) to be executed  
Second parameter indicates the number of milliseconds before execution

Ex.1

```
setTimeout(someFunction, 3000); //using a declared or named function
function someFunction(){
    //something cool happens
}
```

Ex.2

```
setTimeout(function(){           //using an anonymous function
    //something else cool happens
}, 3000)
```

-----

**clearTimeout()** - *Cancels a single time-based execution* if a variable has been set to a `setTimeout()` call  
First parameter is variable set to a `setTimeout()` call

Ex.3

```
var timerExample = setTimeout(someFunction, 3000);
function someFunction(){
    //something cool happens
}
clearTimeout(timerExample);
```

-----

**setInterval()** - *Repeated* execution of code at every given time-interval  
First parameter is a function (callback) to be executed  
Second parameter indicates the number of milliseconds before execution (interval)

Ex.4

```
setInterval(someFunction, 3000); //using a declared or named function
function someFunction(){
    //something cool happens
}
```

Ex.5

```
setInterval(function(){           //using an anonymous function
    //something else cool happens
}, 3000)
```

-----

**clearInterval()** - *Cancels* a repeated execution at every given time-interval if a variable has been set to a `setInterval()` call

Ex.6

```
var timerExample = setInterval(someFunction, 3000);
function someFunction(){
    //something cool happens
}
clearInterval(timerExample);
```