

Callbacks & Schedulers | Cheat Sheet

1. Callback function

A Callback is a function that is passed as an argument to another function.

1.1 Passing a function as an argument

JAVASCRIPT

```
1 function displayGreeting(displayName) {  
2   console.log("Hello");  
3   displayName();  
4   console.log("Good Morning!");  
5 }  
6  
7 displayGreeting(function() {  
8   console.log("Rahul");  
9 });
```

1.2 Passing a function name as an argument

JAVASCRIPT

```
1 function displayGreeting(displayName) {  
2   console.log("Hello");  
3   displayName();  
4   console.log("Good Morning!");  
5 }  
6 function displayRahul() {  
7   console.log("Rahul");  
8 }  
9 displayGreeting(displayRahul);
```

1.3 Passing a function expression as an argument

JAVASCRIPT

```
1 function displayGreeting(displayName) {  
2   console.log("Hello");  
3   displayName();  
4   console.log("Good Morning!");  
5 }  
6 let displayRam = function() {  
7   console.log("Ram");  
8 };  
9 displayGreeting(displayRam);
```

2. Schedulers

The Schedulers are used to schedule the execution of a callback function.

There are different scheduler methods.

- `setInterval()`
- `clearInterval()`
- `setTimeout()`
- `clearTimeout()` , etc.

2.1 `setInterval()`

The

`setInterval()` method allows us to run a function at the specified interval of time repeatedly.

Syntax:

```
setInterval(function, delay);
```

function - a callback function that is called repeatedly at the specified interval of time (

delay). **delay** - time in milliseconds. (1 second = 1000 milliseconds)

JAVASCRIPT

```
1 let counter = 0;
2 setInterval(function() {
3   console.log(counter);
4   counter = counter+1;
5 }, 1000);
```

In the

`setInterval()` method, the callback function repeatedly executes until the browser tab is closed or the scheduler is cancelled.

When we call the

`setInterval()` method, it returns a unique id. This unique Id is used to cancel the callback function execution.

2.2 `clearInterval()`

The

`clearInterval()` method cancels a schedule previously set up by calling `setInterval()` .

To execute

`clearInterval()` method, we need to pass the uniqueId returned by `setInterval()` as an argument.

Syntax:

```
clearInterval(uniqueId);
```

JAVASCRIPT

```
1 let counter = 0;
2 let uniqueId = setInterval(function() {
3   console.log(counter);
4   counter = counter+1;
5 }, 1000);
6
7 clearInterval(uniqueId);
```

Try out the

setInterval() and clearInterval() methods and check the output in the below Code Playground console.

HTML

CSS

JAVASCRIPT

Inspect

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="https://stackp
5     <script src="https://code.jquery.com/jquery-
6     <script src="https://cdn.jsdelivr.net/npm/p
7     <script src="https://stackpath.bootstrapcdn.
8   </head>
9   <body>
10    <button id="setIntervalBtn">
11      Set Interval
12    </button>
13    <button id="clearIntervalBtn">
14      Clear Interval
15    </button>
16  </body>
17 </html>
```

2.3 setTimeout()

The

setTimeout() method executes a function after the specified time.

Syntax:

```
setTimeout(function, delay);
```

function - a callback function that is called after the specified time (

`delay`). **delay** - time in milliseconds.

JAVASCRIPT

```
1 let counter = 0;
2 setTimeout(function() {
3   console.log(counter);
4   counter = counter + 1;
5 }, 1000);
```

2.4 clearTimeout()

We can cancel the

`setTimeout()` before it executes the callback function using the `clearTimeout()` method.

To execute

`clearTimeout()`, we need to pass the `uniqueId` returned by `setTimeout()` as an argument.

Syntax:

`clearTimeout(uniqueId);`

JAVASCRIPT

```
1 let counter = 0;
2 let uniqueId = setTimeout(function() {
3   console.log(counter);
4   counter = counter+1;
5 }, 1000);
6
7 clearTimeout(uniqueId);
```

Try out the

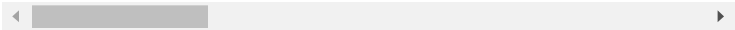
`setTimeout()` and `clearTimeout()` methods and check the output in the below Code Playground console.

HTML CSS JAVASCRIPT

Inspect

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="https://stackpath
5     <script src="https://code.jquery.com/jquery-3.
6     <script src="https://cdn.jsdelivr.net/npm/pop
7     <script src="https://stackpath.bootstrapcdn.co
8   </head>
9   <body>
10    <button id="setTimeoutBtn">
11      Set Timeout
12    </button>
13  </body>
```

```
13     </body>  
14 </html>
```



[Submit Feedback](#)