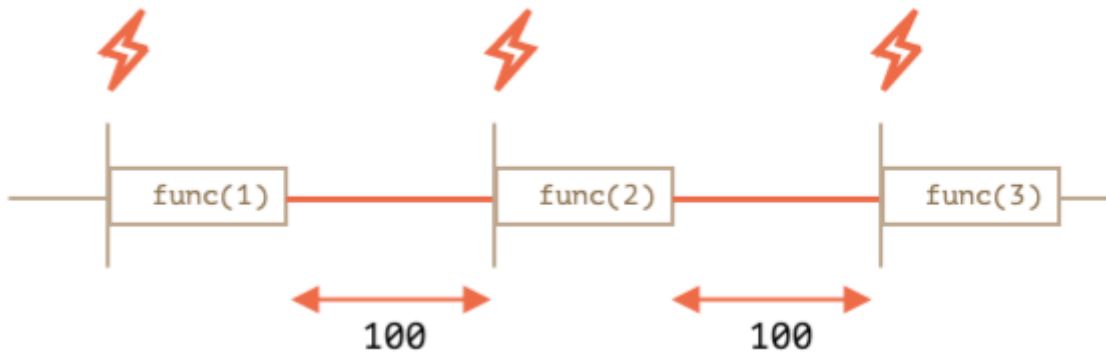


setInterval(callback, delay)



<https://javascript.info/settimeout-setinterval>

1. `setInterval` overview
2. `clearInterval`
3. Example: countdown
4. `setTimeout` and `setInterval`

1. setInterval overview

- What: exec callback every period of time until it's cleared by `clearInterval`.
- When: do something repeatedly
- Where: working on both browser + nodejs

```
const intervalId = setInterval(callback, delay);  
// callback is the function will be executed once the timer is expired  
// delay is the time in milliseconds. 1 second = 1000 milliseconds  
// intervalId is a positive integer, used to identify an interval  
// and clear it by using clearInterval(intervalId)
```

2. clearInterval

```
const intervalId = setInterval(() => {  
  console.log('Hey, are you there?');  
}, 2000);  
clearInterval(intervalId);
```

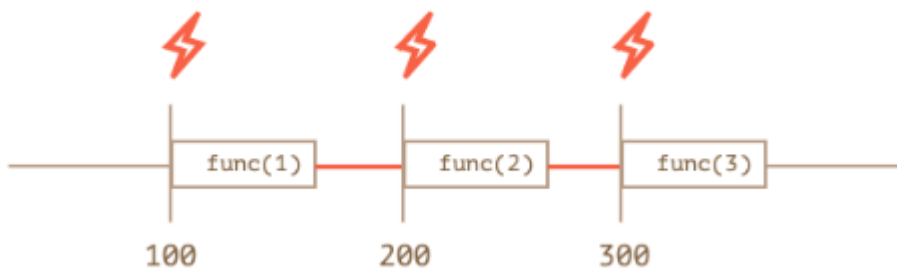
NOTE: `setInterval()` and `setTimeout()` share the same pool of IDs.

3. Example: countdown

```
function countdown(seconds) {  
  let currentSecond = seconds;  
  const intervalId = setInterval(() => {  
    console.log(currentSecond);  
    if (currentSecond <= 0) {  
      clearInterval(intervalId);  
    }  
    currentSecond -= 1;  
  }, 1000);  
}  
countdown(10);
```

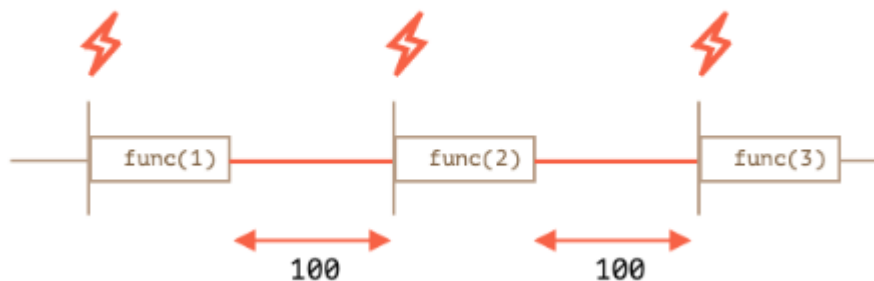
4. setTimeout and setInterval

For `setInterval` the internal scheduler will run `func(i++)` every 100ms:



```
let i = 1;  
setTimeout(function run() {  
  func(i++);  
  setTimeout(run, 100);  
}, 100);
```

And here is the picture for the nested `setTimeout`:



<https://javascript.info/settimeout-setinterval>