

COMP4021  
Internet Computing

# Using Timers

Gibson Lam and David Rossiter

# Using Timers

- You can use the following JavaScript functions to control the exact timing of running some code
  - `setTimeout()`
  - `clearTimeout()`

# setTimeout()

- You can run some code after some time using `setTimeout()`
- For example, you can show an alert message 5 seconds after you run this code:


```
setTimeout(function() {  
    alert("Ring! Ring!");  
}, 5000);
```

*This function  
runs after 5  
seconds*

*5000 milliseconds = 5 seconds*

# Anonymous Functions

- You may notice the function shown in the previous slide does not have a name
- It is called an *anonymous function*

 *No name is inserted here*

```
function() {  
    alert("Ring! Ring!");  
}
```

- This kind of function is commonly used for passing some code as an argument

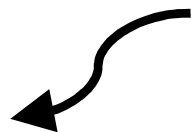
# Cancelling a Timer

- You can cancel a timer using `clearTimeout()`
- To do that, when you set a timer, you need to remember the timer id, like this:

```
let timerid = setTimeout(function() {  
    alert("Ring! Ring!");  
}, 5000);
```

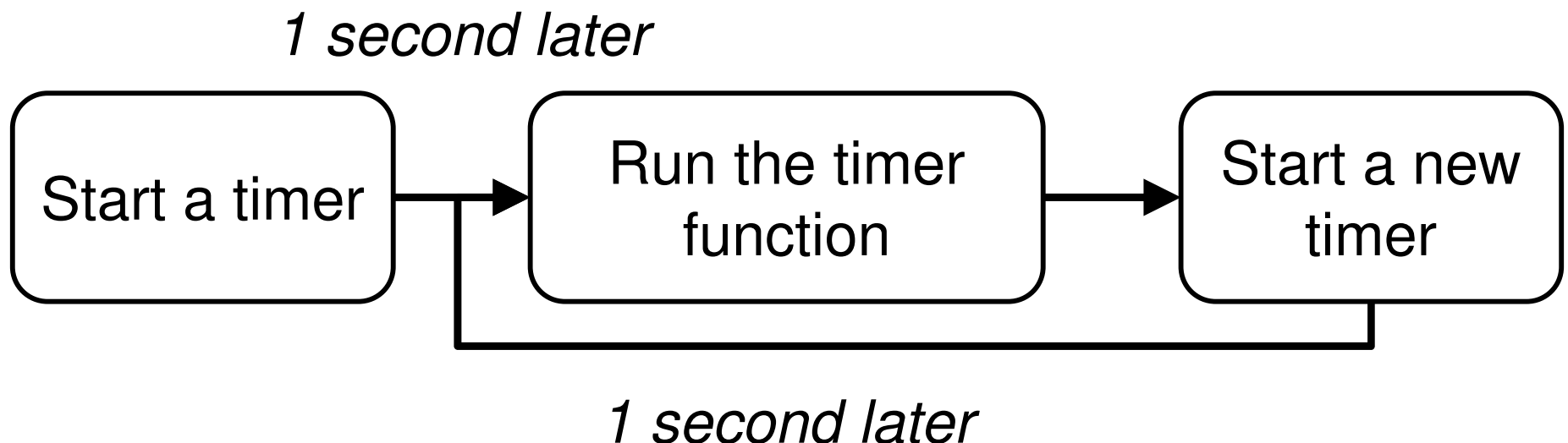
```
clearTimeout(timerid);
```

*No alert message  
is shown after  
running this code*



# Making a Count Down Timer

- Computer games commonly use a count down timer
- You can do that easily using `setTimeout()`
- You just keep making new timers when a previous one fires



# Example of a Countdown Timer

- Here is an example countdown timer that shows a countdown of 5 seconds:

```
let timeRemaining = 5;
                                Update the HTML
                                <h1 id="time">5</h1>
function countDown() {
    timeRemaining = timeRemaining - 1;
    document.getElementById("time")
        .innerHTML = timeRemaining;
    if (timeRemaining > 0)
        setTimeout(countDown, 1000);
}
                                Start the next timeout
                                when the time > 0
setTimeout(countDown, 1000);
```

# Running the Code

