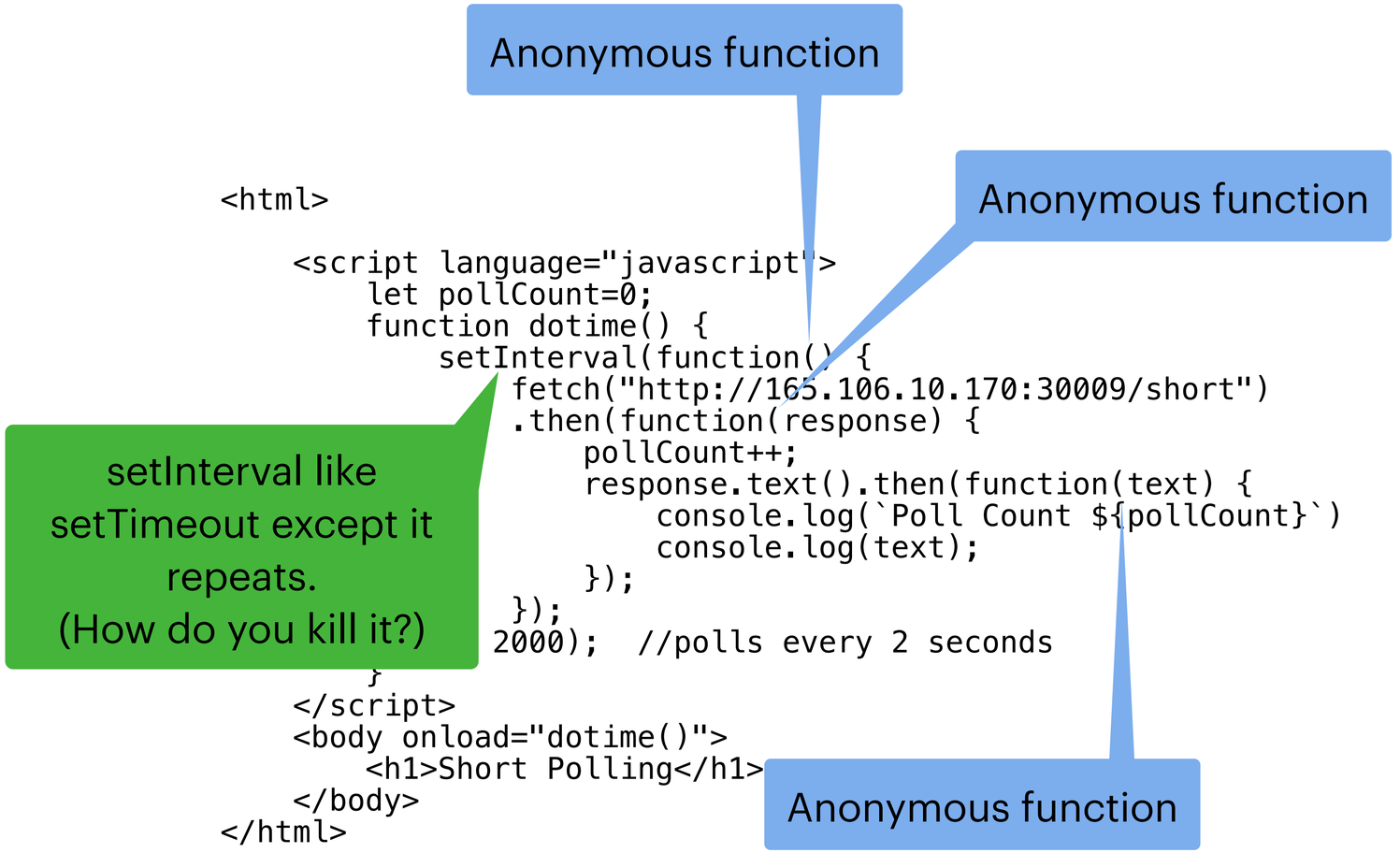
Javascript and SQL

From Monday

- Write a javascript web page to
  - Has an element that changes with every button click js11.html
- Write a javascript web page to
  - Have an element such that on click replace content immediately with 10. After 1 second replace with 9. Do this until 0. (Ie make a countdown timer from 10). js10.html
- Write a javascript web page to
  - Create a span element containing text line "NOT hovering"
    - only when hover over change text to "Stop looking over my shoulder" js9.html

Non-Blocking in Javascript

- Anything that could block must be handled asynchronously
- Generally this means creating an anonymous function containing the things you want to do async
- The anon function gets evaluated at load, but the function is not executed until the async event triggers




```
<html>

    <script language="javascript">
        let pollCount=0;
        function dotime() {
            setInterval(function() {
                fetch("http://165.106.10.170:30009/short")
                .then(function(response) {
                    pollCount++;
                    response.text().then(function(text) {
                        console.log(`Poll Count ${pollCount}`)
                        console.log(text);
                    });
                });
            2000);  //polls every 2 seconds
        }
    </script>
    <body onload="dotime()">
        <h1>Short Polling</h1>
    </body>
</html>
```

Most languages have blocking constructs

- Most languages have some form of parallel execution
  - "Thread"
  - So even if one thread is blocked thing can keep happening
    - For example, a simple hello world server in Go with a 15 second sleep block.
    - Problems with using threads to solve blocking?

- Android / iOS both disallow blocking in the main IO thread

```go
package main

import (
    "fmt"
    "net/http"
    "time"
)


func hw(w http.ResponseWriter, req *http.Request) {
    fmt.Println("Enter hello")
    time.Sleep(15*time.Second)
    fmt.Fprintf(w, "hello world")
    fmt.Println("Exit hello")
}


func main() {
    http.HandleFunc("/hello", hw)
    http.ListenAndServe(":30030", nil)
}
```

Explain what happens, Why?
when click on the span1 element

```
<html>
    <body>
        <script>

            function ra() {
                let ii=23;
                setTimeout( function() {

document.querySelector("#span1").innerHTML=`${ii}`;
                }, 5000)
                ii += 19;
            }
        </script>
        <span onclick="ra();" id="span1"
style="font-size:300%;color:red">Original 1</span>
        <br/>
    </body>
</html>
```

If this is not what you want,
how do you change code?

```
<html>
    <body>
        <script>
            function funny(fnc) {
                setTimeout(function()
{ fnc() }, 2000);
            }

            function ra(argg) {
                let ii=23;
                let gg = function() {

document.querySelector("#span1").innerHTML=`$
{ii} ${argg}`;
                argg+=30;
                };
               funny(gg);
                ii += 19;
            }
        </script>
        <span onclick="ra(12);" id="span1"
style="font-size:300%;color:red">Original 1</
span>
        <br/>
    </body>
</html>
```