HTML 5
Web Workers

Bok, Jong Soon
javaexpert@nate.com
https://github.com/swacademy/HTML5

What is a Web Worker?

- When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.
- Is a JavaScript that runs
  - In the background,
  - Independently of other scripts
  - Without affecting the performance of the page.

Check Web Worker Support

- Before creating a web worker, check whether the user's browser supports it.

```
1  if(typeof(Worker)!=="undefined") {
2    // Yes! Web worker support!
3    // Some code.....
4  } else {
5    // Sorry! No Web Worker support..
6  }
```

Create a Web Worker File

- The `postMessage()` method
  - Is used to posts a message back to the HTML page.

```
1  var i=0;
2
3  function timedCount() {
4      i=i+1;
5      postMessage(i);
6      setTimeout("timedCount()",500);
7  }
8  timedCount();
9
```

Create a Web Worker Object

- We need to call web “worker file” from an HTML page.
- The following lines checks if the worker already exists, if not - it creates a new web worker object and runs the code in "demo_workers.js":

```
11  function startWorker() {
12      if(typeof(Worker)!=="undefined") {
13          if(typeof(w)=="undefined") {
14              w=new Worker("demo_workers.js");
15          }
16          w.onmessage = function (event) {
```

Create a Web Worker Object (Cont.)

- Then we can send and receive messages from the web worker.
- Add an **`onmessage`** event listener to the web worker.

```
15          }
16          w.onmessage = function (event) {
17              document.getElementById("result").innerHTML=event.data;
18          };
19      } else {
```

- When the web worker posts a message, the code within the event listener is executed.
- The data from the web worker is stored in **`event.data`**.

Terminate a Web Worker

- When a web worker object is created, it will continue to listen for messages (even after the external script is finished) until it is terminated.
- To terminate a web worker, and free browser/computer resources, use the `terminate()` method:

```
w.terminate();
```

Lab1 : Web Worker

- Web Browsers
  - Edge, Firefox, Google Chrome, Opera, Safari
- Text Editors
  - Visual Studio Code, Notepad++ or Editplus, etc…
- Files
  - webworker.html
  - demo_workers.js

Lab1 : webworker.html

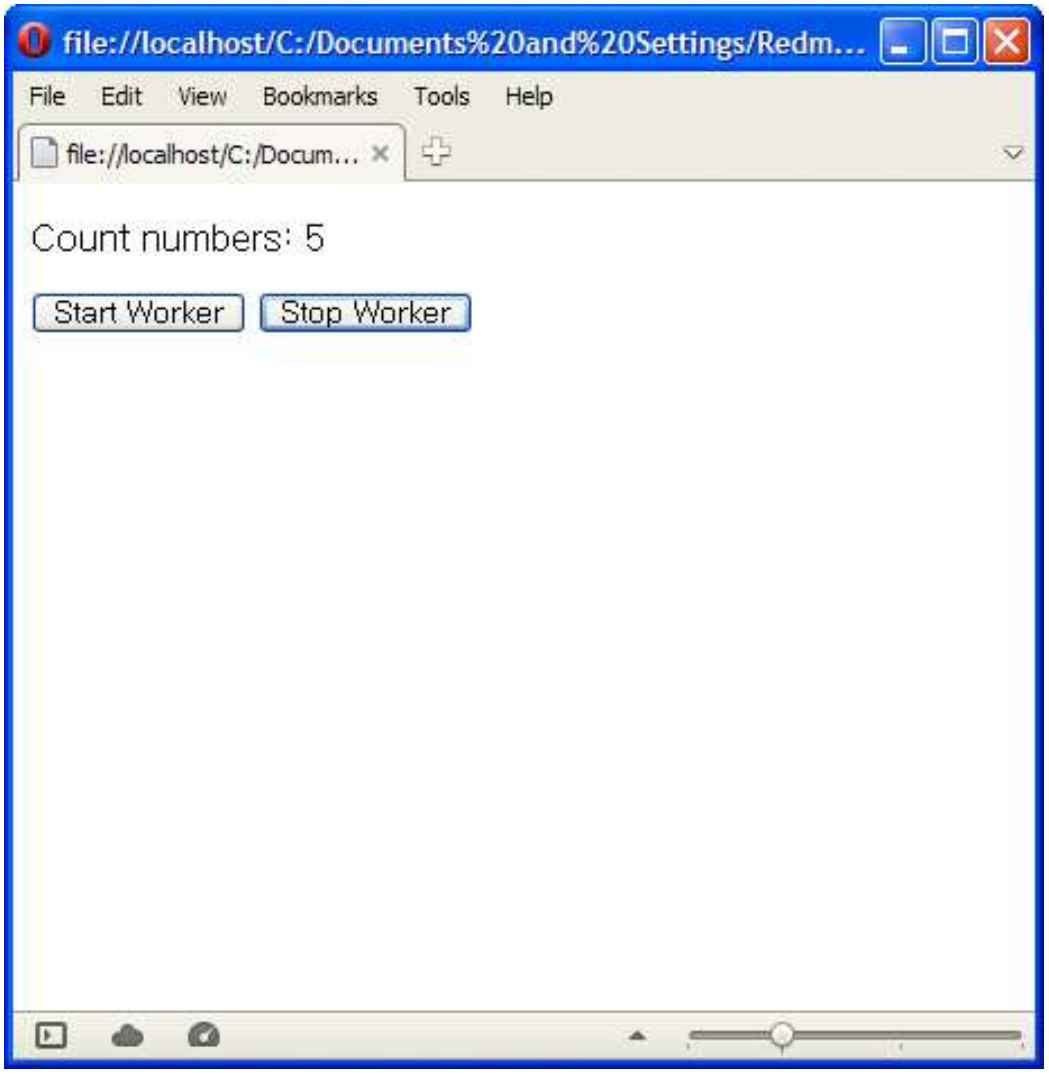
```
<!DOCTYPE html>
<html>
<body>
<p>Count numbers: <output id="result"></output></p>
<button onclick="startWorker()">Start Worker</button>
<button onclick="stopWorker()">Stop Worker</button>
<br><br>
<script>
var w;

function startWorker() {
if(typeof(Worker)!=="undefined") {
if(typeof(w)=="undefined") {
w=new Worker("demo_workers.js");
}
w.onmessage = function (event) {
document.getElementById("result").innerHTML=event.data;
};
} else {
document.getElementById("result").innerHTML=
"Sorry, your browser does not support Web Workers...";
}
}
function stopWorker() {
w.terminate();
}
</script>
</body>
</html>
```

Lab1 : demo_workers.js

```
1  var i=0;
2
3  function timedCount() {
4      i=i+1;
5      postMessage(i);
6      setTimeout("timedCount()",500);
7  }
8  timedCount();
9
```

Lab1 : Result


file://localhost/C:/Documents%20and%20Settings/Redm...
File Edit View Bookmarks Tools Help
file://localhost/C:/Docum...
Count numbers: 5
Start Worker
Stop Worker