

Web Worker

When executing scripts in an HTML page, the page becomes unresponsive until the script is finished.

A web worker is a JavaScript that runs in the background, independently of other scripts, without affecting the performance of the page. You can continue to do whatever you want: clicking, selecting things, etc., while the web worker runs in the background.

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":

```
var w = new Worker("demo_workers.js");  
if(typeof(w) == "undefined") {  
}
```

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.

```
w.onmessage = function(event){  
    document.getElementById("result").innerHTML = event.data;  
};
```

Demo.html

```
<!DOCTYPE html>  
<html>  
<body>  
<p>Count: <output id="result"></output></p>  
<button onclick="startWorker()">Start Counter</button>  
<button onclick="stopWorker()">Stop Counter</button>  
<script>  
    var w;  
    function startWorker() {  
        if (typeof (Worker) != "undefined") {  
            if (typeof (w) == "undefined") {  
                w = new Worker("Demo.js");  
            }  
            w.onmessage = function (event) {  
                document.getElementById("result").innerHTML = event.data;  
            }  
        }  
    }  
}
```

```
    };  
  }  
  else {  
    document.getElementById("result").innerHTML = "This browser does not support Web Workers";  
  }  
}  
  
function stopWorker() {  
  w.terminate();  
}  
  
</script>  
</body>  
</html>
```

Demo.js

```
var i = 0;  
  
function timedCount() {  
  i = i + 1;  
  postMessage(i);  
  setTimeout("timedCount()", 500);  
}  
timedCount();
```

Note: Since web workers are in external files, they do not have access to the following JavaScript objects:

- The window object
- The document object
- The parent object