

Web Workers

I **Web Workers** sono delle API (*Application Programming Interface*) che permettono l'esecuzione in background e in modo asincrono di script che richiedono lunghe elaborazioni, senza interferire con la navigazione dell'utente.

Il loro uso è importante nel caso di esecuzione di codice JavaScript molto complessi che potrebbero causare una lunga attesa del browser, compromettendo l'efficienza della navigazione.

La modalità di lavoro dei Web Workers è la seguente: viene creato un nuovo *thread* di elaborazione all'interno del quale si esegue lo script, superando le precedenti difficoltà di comunicazione tra i vari *thread*, e fornendo delle funzioni che hanno lo scopo di far passare i dati da un *thread* (la pagina Web vera e propria) all'altro (i vari Web Workers).

I Web Workers sono normalmente contenuti in file *.js* esterni e richiamati all'interno della pagina. Per questioni di sicurezza, i Web Workers non hanno la possibilità di accedere agli oggetti *window* e *document*, pertanto non sono in grado di utilizzare le API per la gestione del DOM della pagina, e quindi di modificarla.

Ciò non significa che non sia possibile interagire con essi, ma è necessario utilizzare gli eventi per poter passare le informazioni necessarie: essi possono essere sia *monodirezionali*, dal WebWorker alla pagina o viceversa, o *bidirezionali*, quando i dati scambiati provengono da entrambe le fonti.

I metodi utilizzati per la comunicazione con i Web Workers sono:

`postMessage()`
`onmessage()`.

Tutte le comunicazioni vengono gestite tramite messaggi attraverso il metodo **`postMessage`**, quando è necessario comunicare tra il WebWorker e la pagina. La ricezione delle informazioni viene gestita dall'evento **`onmessage`**.

La creazione di un Web Worker avviene attraverso la seguente istruzione:

```
var worker = new Worker('mioworker.js');
```

Si noti la presenza del file esterno *mioworker.js*, nel quale è presente il codice da eseguire.

Lo script *mioworker.js* viene eseguito in asincrono e può inviare dati alla pagina Web grazie al metodo *postMessage*.

Quindi se il file *mioworker.js* contiene:

```
postMessage('Quanti anni hai?');
```

nella pagina Web ci sono le seguenti istruzioni:

```
var worker = new Worker('mioworker.js');  
mioworker.onmessage = function (event) {  
  mioworker.postMessage(prompt(event.data));  
}
```