



JavaScript: monitorare le modifiche al DOM

Possiamo monitorare i cambiamenti apportati al DOM durante il ciclo vitale di un documento utilizzando l'oggetto `MutationObserver`. Questo oggetto sostituisce i Mutation Events definiti in precedenza nelle specifiche DOM3 Events.

L'oggetto `MutationObserver` viene per prima cosa istanziato con una funzione di callback che accetta come parametro un array di oggetti `MutationRecord` che memorizzano le mutazioni avvenute in un elemento scelto come target:

```
const target = document.getElementById( 'test' );

const observer = new MutationObserver(function(
mutations ) {
  mutations.forEach( mutation => {
    console.log( mutation.type, mutation.target,
mutation.attributeName );
  });
});
```

`type`, `target` e `attributeName` sono proprietà dell'oggetto `MutationRecord`: la prima restituisce una stringa contenente il nome della mutazione avvenuta, la seconda è un riferimento all'elemento scelto

come target mentre la terza restituisce il nome locale dell'attributo modificato.

A questo punto occorre inizializzare l'oggetto passandogli dei parametri di configurazione:

```
const config = { attributes: true };  
  
observer.observe( target, config );
```

In questo caso stiamo monitorando i cambiamenti avvenuti negli attributi dell'elemento. In seguito possiamo interrompere il monitoraggio in questo modo:

```
observer.disconnect();
```

Questo oggetto si rivela molto utile quando dobbiamo eseguire delle azioni in base ai cambiamenti apportati al documento, ad esempio dopo che l'utente ha eseguito una determinata azione sulla pagina.