Javascript – Object destructuring

Object destructuring

Nell'utilizzo del JavaScript, una delle strutture dati più utilizzata è l'oggetto. Il modo più semplice per recuperare le sue proprietà, è quello di accedervi seguendo la "dot notation", come in questo esempio:

```
var obj = { nome: 'Foo', cognome: 'Bar',
}; var nome = obj.nome;
var cognome = obj.cognome;
```

Se avessimo un oggetto con 5 proprietà, dovremmo ripetere l'operazione per accedere ad ognuna di essa. Certamente, una prima ottimizzazione (in termini di righe di codice), sarebbe quella di dichiarare le variabili su una sola riga

```
var nome = obj.nome, cognome = obj.cognome;
```

Quello che salta all'occhio è come il suffisso *obj.* si ripete ad ogni dichiarazione.

E' proprio qui che entra in gioco **l'object destructuring,** il quale ci consente di dichiarare una variabile e nello stesso momento recuperarne il contenuto, in questo modo:

```
var { nome, cognome } = obj;
```

La sintassi può sembrare un pò strana, ma quello che fa è veramente semplice: dichiara e inizializza due variabili **nome** e **cognome** proprietà dell'oggetto **obj** e ne assegna il valore.

Fin qui tutto bene, ma veniamo al punto dove questa sintassi rende le cose molto più snelle e facili da leggere: passaggio di valori ad una funzione.

Supponiamo di avere una funzione **somma** la quale prende come input due valori e ne restituisce la somma, questa potrebbe essere dichiarata semplicemente scrivendo:

```
function somma(a, b) { return a + b;
}
```

E se le proprietà **a** e **b** fossero contenute in un oggetto? Semplice, utilizzando **l'object destructuring** possiamo definire le variabili nella firma della funzione e contestualmente estrarne i valori:

```
function somma({ a, b }) { return a + b;
} var obj = { a: 10, b: 20 }; somma(obj);
```

Ovviamente se le variabili non provenissero da un oggetto, potremmo sempre dichiararne uno "inline" e quindi invocare la funzione sempre nello stesso modo:

```
var dieci = 10;
var venti = 20; somma({ a: dieci, b: venti });
```

In questo esempio, le variabili sono state volutamente chiamate con nomi diversi rispetto alle proprietà dell'oggetto, altrimenti se fossero state chiamate **a** e **b** il risultato sarebbe stato:

```
somma({ a: a, b: b });
```

Questo però è quasi sempre lo scenario più comune, dove una proprietà si chiama esattamente come il nome della sua variabile. Se cosi fosse, potremmo omettere la parte successiva ai "due punti" e questa verrà automaticamente trasformata in dichiarazione/valore:

```
// somma({ a: a, b: b }) questo si può abbreviare in.. somma({
a, b });
```

https://italiancoders.it/javascript-object-destructuring/