

INTRO

Un oggetto è un contenitore di valori eterogenei, messi insieme a formare una struttura dati unica e tale da avere generalmente una particolare identità.

Normalmente, infatti, un oggetto è utilizzato per rappresentare un'entità specifica come ad esempio una persona, un ordine, una fattura, una prenotazione, etc. tramite un'aggregazione di dati e di funzionalità.

Un oggetto tipicamente possiede:

- Dati, detti proprietà e rappresentati da coppie di nome-valore;
- Funzionalità, sono dette metodi e rappresentate da funzioni.

Object literal

Un modo per creare un oggetto è la rappresentazione letterale: si potrebbe dire che definiamo l'oggetto a partire dal proprio contenuto.

```
var oggettoVuoto = {};
```

```
var persona = { "nome": "Mario", "cognome": "Rossi" };
```

Tramite la notazione letterale rappresentiamo un oggetto racchiudendo tra parentesi graffe le sue proprietà e i suoi metodi.

Nella prima dichiarazione abbiamo creato un oggetto vuoto, cioè un oggetto senza proprietà né metodi, tutto sommato non di grande utilità, ma importante per capire che un oggetto è in fondo una lista di coppie di valori e come ogni lista può essere vuota.

Infatti, come possiamo vedere nella seconda dichiarazione, per definire l'oggetto persona abbiamo elencato due coppie di stringhe separate da virgole. Ciascuna coppia è formata da due stringhe separate da due punti. La prima stringa rappresenta il nome della proprietà mentre la seconda indica il suo valore.

Proprietà

Per i nomi delle proprietà di un oggetto non abbiamo le restrizioni dei nomi delle variabili. Per essi può essere utilizzata una qualsiasi stringa, anche se con qualche vincolo quando tali proprietà vengono richiamate. I doppi apici intorno al nome della proprietà sono opzionali, ma diventano obbligatori quando il nome non segue le regole per nomi delle variabili. Quindi, avremmo potuto scrivere la nostra definizione di persona nel seguente modo:

```
var persona = { nome: "Mario", cognome: "Rossi"};
```

Una proprietà può assumere qualsiasi valore derivante da un'espressione JavaScript, compreso un altro oggetto. Quindi possiamo creare oggetti annidati come mostrato di seguito:

```
var persona = {  
    nome: "Mario",  
    cognome: "Rossi",  
    indirizzo: {  
        via: "Via Garibaldi",  
        numero: 15,  
        CAP: "00100",  
        città: "Roma"  
    }  
};
```

Per accedere ai valori memorizzati in una proprietà di un oggetto abbiamo due approcci. Il primo approccio è la cosiddetta dot-notation in base alla quale indichiamo un oggetto e la proprietà a cui siamo interessati separandoli con un punto:

```
var nome = persona.nome;
```