

Introduzione alla programmazione orientata agli oggetti

Oggetti, metodi e proprietà

Nelle seguenti pagine, non solo si conosceranno nuovi concetti e nuovi termini, ma si vedranno anche alcune utili funzioni di output in Javascript.

Metodi

I **metodi** non sono niente di nuovo. È un altro nome per chiamare quelle che sono chiamate anche funzioni. Sulle funzioni non si deve dire molto perchè abbiamo scritto diversi programmi dove sono state create diverse funzioni.

Un metodo, o una funzione, permette di eseguire un elenco di istruzioni e si può riconoscere dal fatto che il suo nome termina con le parentesi tonde.

Nel seguente esempio Javascript vengono create due funzioni: azione1() e azione2().

Il programma Javascript completo si trova in fondo a questo documento.

```
7      function azione1()
8      {
9          window.alert("avviso in nuova finestra");
10         document.write("messaggio in un nuovo documento html");
11     }
12
13     function azione2()
14     {
15         document.getElementById("secondo").innerHTML="sorpresa";
16     }
```

Esaminiamo la riga numero 9 perché è un tipo di istruzione già nota:

```
window.alert("avviso in nuova finestra");
```

Questa istruzione si può scomporre in diverse parti: ogni parte ha un nome diverso a seconda della posizione che occupa, come nel seguente modello:

```
oggetto.metodo(argomento);
```

In questo caso, si può dire allora che:

- **l'oggetto** è `window` cioè la finestra del browser
 - **il metodo** (o funzione) è `alert()`, che viene applicato all'oggetto `window`
 - **l'argomento** è la frase tra virgolette `"avviso in nuova finestra"` che viene passato al metodo
- Le parole in grassetto sono dei termini specifici, ma a volte possono essere usati dei sinonimi: "metodo" è sinonimo di "funzione" e "argomento" è sinonimo di "parametro". I verbi che sono sottolineati non sono dei termini specifici, ma per evitare di aggiungere troppe parole nuove si utilizzeranno solo quelli.

Come si legge il codice?

Chi sa leggere il codice riesce a comprendere anche il significato delle istruzioni.

Quando si vuole comprendere il significato di un'istruzione che ha la seguente forma

```
oggetto.metodo(argomento);
```

si potrebbe immaginare di leggerla come la seguente frase:

- Un oggetto esegue un metodo che prende un argomento (tra parentesi).

oppure (leggendola in direzione inversa)

- Un argomento viene passato ad un metodo che viene applicata su un oggetto.

Dopo aver detto questo, come si potrebbe leggere la riga numero 9?

L'oggetto `window` esegue il metodo `alert()` che prende l'argomento: `"avviso in nuova finestra"`.

Esercizio

Ora si può provare a leggere da soli questa istruzione (vedi riga 10 del precedente programma) e poi confrontarla con la soluzione proposta:

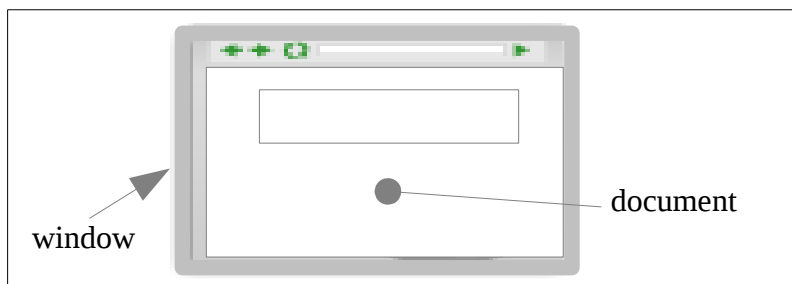
```
document.write("messaggio in un nuovo documento html");
```

- l'oggetto `document` esegue il metodo `write()` che prende l'argomento `"messaggio..."`
oppure (leggendola in direzione inversa)
- l'argomento `"messaggio..."` viene passato al metodo `write()` che viene applicato sull'oggetto `document`.

Per capire meglio è necessario provare il programma che si trova in fondo a questo documento.

Oggetti

Se prima abbiamo detto che `window` è un oggetto che rappresenta la finestra del browser, allora in modo simile, `document` è l'oggetto contenuto in questa finestra. Questi oggetti ci sembrano visibili e reali, come le finestre, ma nella memoria del computer sono rappresentati contenitori di numeri, quindi gli oggetti sono molto simili ad un insieme di "grandi" variabili.



Qualsiasi cosa reale o astratta può essere rappresentata nella memoria del computer come un contenitore di numeri, ad esempio:

- la finestra del browser sul monitor può essere rappresentata memorizzando la dimensione della base e dell'altezza e la sua posizione attuale
- una persona potrebbe essere rappresentata con i suoi dati: nome, altezza, peso, numero di telefono, e codice fiscale
- un motorino potrebbe essere rappresentato con: targa, cilindrata, colore, peso.

Come si è già visto, un oggetto può eseguire metodi (cioè funzioni), oppure, invertendo la frase, un metodo può essere applicato su un oggetto. Nei linguaggi di programmazione orientati agli oggetti, i metodi e gli oggetti sono strettamente legati e non si può utilizzare uno specifico metodo senza il suo corrispondente oggetto. Ad esempio, non si può usare `write()` senza l'oggetto `document`.

Come paragone possiamo immaginare un oggetto `motorino` e un metodo `frenare()`: non si può usare il metodo `frenare` senza l'oggetto `motorino`.

Adesso vedremo anche che un oggetto possiede una proprietà (cioè delle caratteristiche), oppure, invertendo la frase, che una proprietà appartiene ad un oggetto.

Proprietà

Prima di continuare a leggere le istruzioni del programma è necessario conoscere un nuovo termine della programmazione orientata agli oggetti. Le proprietà non sono una novità. Le abbiamo già usate quando abbiamo scritto

```
var x = document.forms["esempio"]["txt_x"].value;
```

Questa istruzione è un'assegnazione, cioè memorizza un valore dentro la variabile x.

- `value` significa "il valore di"...
- `forms[...]["txt_x"]` specifica il campo input "txt_x" contenuto nel form
- `document` è un'oggetto che rappresenta il codice html della pagina attualmente aperta nella finestra del browser

`value` non ha le parentesi tonde come un metodo, quindi non è un metodo, ma indica una proprietà (una caratteristica di un oggetto). In questo caso la proprietà è il valore inserito.

Si può leggere dicendo che:

- l'oggetto `document` possiede la proprietà `forms[...]` che, essendo a sua volta un oggetto, possiede la proprietà `value`
- la proprietà `value` appartiene all'oggetto `forms[...]`, che a sua volta è una proprietà che appartiene all'oggetto `document`

Esercizio

studiare sul manuale di riferimento il metodo `open` (appartiene all'oggetto `window`)

<http://www.w3schools.com/jsref/default.asp>

PROMEMORIA: prima di andare avanti si deve studiare che un metodo può restituire qualcosa...

A questo punto si può analizzare la riga 15 del precedente programma:

```
document.getElementById("secondo").innerHTML = "sorpresa";
```

si tratta di un'assegnazione, cioè si deve memorizzare la parte a sinistra ("sorpresa"), nella parte di destra (come nell'istruzione `x = 5;`). Poiché l'assegnazione dovrebbe essere già chiara, concentriamo l'attenzione solo sulla parte a sinistra del segno uguale, che si potrebbe leggere come:

- l'oggetto `document` esegue il metodo `getElementById()`, e poi questo metodo, restituendo un oggetto, possiede la proprietà `innerHTML`
oppure (leggendola in direzione inversa)
- la proprietà `innerHTML` appartiene all'oggetto restituito dal metodo `getElementById()`, che viene applicato sull'oggetto `document`
- argomento `"messaggio..."` viene passato al metodo `write()` che viene applicato sull'oggetto `document`.

Per capire meglio è necessario provare il codice che si trova in fondo a questo documento.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8">
5      <title>oggetto document</title>
6      <script type="text/javascript">
7        function azione1()
8        {
9          window.alert("avviso in nuova finestra");
10         document.write("messaggio in un nuovo documento html");
11        }
12
13        function azione2()
14        {
15          document.getElementById("secondo").innerHTML="sorpresa";
16        }
17      </script>
18    </head>
19    <body>
20      <h1>window è un oggetto: è la finestra del browser</h1>
21      <p>Questa pagina esegue l'istruzione<p>
22      <pre>window.alert() </pre>
23      <p>cioè un'azione sull'oggetto window<p>
24      <h1 id="secondo">document è un oggetto: è il documento contenuto nel browser</h1>
25      <p>La pagina esegue poi l'istruzione<p>
26      <pre>document.write() </pre>
27      <p>cioè un'azione sull'oggetto document</p>
28      <form>
29        <input type="button" value="azione1" onclick="azione1()">
30        <input type="button" value="azione2" onclick="azione2()">
31      </form>
32    </body>
33  </html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>oggetto document</title>
    <script type="text/javascript">
      function azione1()
      {
        window.alert("avviso in nuova finestra");
        document.write("messaggio in un nuovo documento html");
      }

      function azione2()
      {
        document.getElementById("secondo").innerHTML="sorpresa";
      }
    </script>
  </head>
  <body>
    <h1>window è un oggetto: è la finestra del browser</h1>
    <p>Questa pagina esegue l'istruzione<p>
    <pre>window.alert() </pre>
    <p>cioè un'azione sull'oggetto window<p>
    <h1 id="secondo">document è un oggetto: è il documento contenuto nel
browser</h1>
    <p>La pagina esegue poi l'istruzione</p>
    <pre>document.write() </pre>
    <p>cioè un'azione sull'oggetto document</p>
    <form>
      <input type="button" value="azione1" onclick="azione1()">
      <input type="button" value="azione2" onclick="azione2()">
    </form>
  </body>
</html>
```