Document Object Model

Document Object Model

- API per documenti html e xml
  - Non è una particolare applicazione o prodotto
  - È una interfaccia che i browser devono implementare per conformarsi allo standard W3C DOM
- Per uno sviluppatore, significa due cose
  - Fornisce una rappresentazione strutturata del documento
  - Definisce come accedere a questa struttura da script, permettendo di gestire le pagine web come un gruppo strutturato di nodi
- Essenzialmente, connette le pagine web agli script e ai linguaggi di programmazione

Esempio

- Es. tutti i browser che implementano DOM devono restituire tutti gli elementi <p> in una pagina HTML come array di nodi quando viene invocato il metodo *getElementsByTagName* del documento:

```
paragrafi = document.getElementsByTagName("p");
// paragrafi[0] è il primo elemento <p>
// paragrafi[1] è il secondo elemento <p>
ecc.
alert(paragrafi[0].nodeName);
```

Oggetto Window

- Proprietà
  - ***document*** – Oggetto documento
  - ***event*** – Evento attuale
  - ***location*** – Url attuale
  - ***name*** – Nome della finestra
  - ***navigator*** – Oggetto navigator (caratteristiche del browser)
  - ***self, parent, top*** – Frame attuale, genitore o radice
  - ***status*** – Messaggio della barra di stato
- Metodi
  - ***alert(msg), confirm(msg), prompt(msg)*** – Visualizza una finestra di dialogo
  - ***open(url, name, ...), close()*** – Apre o chiude una finestra
  - ***setTimeout(expr, millis)*** – Valuta una espressione dopo un intervallo di tempo specificato

Oggetto Navigator

- Informazioni sul browser usato dall'utente
- Proprietà
  - ***appName*** – Nome del browser
  - ***appVersion*** – Piattaforma e versione del browser
  - ***browserLanguage*** – Lingua del browser
  - ***cookieEnabled*** – Cookie abilitati, o no?
  - ***cpuClass*** – Stringa che identifica la classe della CPU
  - ***onLine*** – Il sistema è on-line, o no?
  - ***platform*** – Piattaforma del browser
  - ***systemLanguage*** – Lingua di default del sistema
  - ***userAgent*** – Agente-utente HTTP
  - ***userLanguage*** – Attuale lingua impostata dall'utente

Oggetto Document

- Documento html contenuto nella finestra
- Proprietà
  - ***anchors, applets, forms, images, links*** – Collezione di tutti gli elementi anchor (...) del documento
  - ***cookie*** – Cookies del documento
  - ***body*** – Elemento body o frameset
  - ***title*** – Titolo del documento
- Metodi
  - ***open(), close()*** – Apre o chiude un documento
  - ***write(text)*** – Scrive testo su un documento
  - ***getElementById(id)*** – Trova un elemento dato il suo id
  - ***createElement(type)*** – Crea un elemento del tipo specificato

Element

- Tutti gli elementi condividono una interfaccia comune
  - Ci sono interfacce più specializzate per oggetti particolari
  - L'elemento *body*, per esempio, ha funzioni e proprietà extra
- Proprietà
  - ***childNodes*** – Array dei nodi figlio dell'elemento
  - ***innerHTML*** – Tutto il contenuto, con il markup, all'interno di un dato elemento
  - ***style*** – Il blocco di regole di stile per l'elemento corrente
- Metodi
  - ***appendChild(element)*** – Aggiunge il nodo specificato nella lista di nodi del documento attuale
  - ***getElementsByTagName(name)*** – Restituisce la collezione di tutti gli elementi con uno specifico nome di tag
  - ***getAttribute(name), setAttribute(name, value)*** – Restituisce o modifica un attributo dell'elemento

Oggetto Form

- Proprietà
  - ***action*** – Attributo action del form
  - ***elements*** – collezione degli elementi del form
  - ***length*** – Numero di elementi del form
  - ***method*** – Metodo http per sottomettere il form
  - ***name*** – Nome del form
  - ***target*** – Frame o finestra dove visualizzare la risposta
- Metodi
  - ***reset()*** – Cancella i valori inseriti dall'utente nel form
  - ***submit()*** – Sottomette il form

Esempio
Validare un form

```
<html>
<head>
<script language="JavaScript">
function validate() {
  campo = document.getElementById('id1');
  if (campo.value.length > 0) {
    return true;
  }
  else {
    alert('Campo vuoto!');
    return false;
  }
}
</script>
</head>
<body>
<form name="form1" onsubmit="return validate()">
<input id="id1" name="field1" type="text" />
<input type="submit" />
</form>
</body>
</html>
```

Esempio
Nascondere un elemento

```
<html>
<head>
<script language="JavaScript">
function hide(elm) {
  document.getElementById(elm).style.visibility = 'hidden';
}
function show(elm) {
  document.getElementById(elm).style.visibility = 'visible';
}
</script>
</head>
<body>
<div id="id1" onclick="hide('id3')">Nascondi</div>
<div id="id2" onclick="show('id3')">Visualizza</div>
<div id="id3">Testo visibile o invisibile</div>
</body>
</html>
```

Esempio
Muovere un elemento

```
<html>
<head>
<script language="JavaScript">
  var t = null; var x = 0;
  function move() {
    x += 5; if (x > 500) x = 0;
    document.getElementById('id3').style.left = x + 'px';
    t = window.setTimeout('move()', 500);
  }
  function home() {
    window.clearTimeout(t); x = 0; t = null;
    document.getElementById('id3').style.left = x + 'px';
  }
</script>
</head>
<body>
  <div id="id1" onclick="if (t == null) move()">Move</div>
  <div id="id2" onclick="home()">Home</div>
  <div id="id3" style="position:absolute;left:0;">Some text</div>
</body>
</html>
```

Esempi
Creare un elemento

```
<html>
<head>
<script language="JavaScript">
function create() {
  var id2 = document.getElementById('id2');
  id2.innerHTML = 'Enter your <b>password</b>: ';
  var newElement = document.createElement('input');
  newElement.setAttribute('type', 'password');
  id2.appendChild(newElement);
}
</script>
</head>
<body>
<div id="id1" onclick="create()">Create</div>
<form id="id2">Some text</form>
</body>
</html>
```